HIGH-ORDER MAXIMUM PRINCIPLE PRESERVING (MPP) TECHNIQUES FOR SOLVING CONSERVATION LAWS WITH APPLICATIONS ON MULTIPHASE FLOW

A Dissertation

by

MANUEL QUEZADA DE LUNA

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Jean-Luc Guermond
Co-Chair of Committee,	Meinhard T. Schobeiri
Committee Members,	Wolfgang Bangerth
	Bojan Popov
Head of Department,	Emil Straube

August 2016

Major Subject: Mathematics

ABSTRACT

We develop numerical methods to solve the linear scalar conservation law fulfilling the maximum principle. To do this we use continuous and discontinuous Galerkin finite elements and achieve the preservation on the maximum principle via the Flux Corrected Transport (FCT) method. We use high-order polynomial spaces with Bernstein basis functions and obtain the optimal convergence rates with spaces of up to third order for smooth solutions that are monotone. This methodology produces good quality results for spaces up to (around) third order. However, when higher-order spaces are used non-physical oscillations are introduced, which is true nevertheless the methods are maximum principle preserving. These oscillations can be highly reduced by defining tighter bounds. Using discontinuous Galerkin finite elements we present a new FCT-*like* methodology based on single cell flux corrections. This method combines a mass conservative low-order Maximum Principle Preserving (MPP) solution with a non mass conservative high-order MPP solution. The process is designed to recover mass conservation locally (with respect to degrees of freedom). Using this scheme we obtain the optimal convergence rates with spaces of up to third order for smooth solutions that are monotone. The method is designed to overcome problems when high-order spaces are used and, under this context, we obtained better results than with the standard FCT method. We present two methods to transport a smoothed Heaviside level set function using a one-stage reinitialization based on artificial compression. The first method allows arbitrarily large compression which might lead to non-physical behavior. To over come this difficulty the second method self balances the artificial dissipation and compression. Finally, we use the level set solver with a Navier-Stokes solver to simulate incompressible two-phase flow.

ACKNOWLEDGEMENTS

Having finished my PhD gives me yet another reason to be thankful and another opportunity in my life to realize and acknowledge the importance and influence of so many people in the construction of my professional and personal character. It is now the moment to humbly acknowledge the people that have influenced and helped me through this process and to express the best I can and in few lines how grateful I am.

First I want to thank my family for their continuous support. Just for being there in good and (maybe more importantly) bad times. Related to this it is impossible not to mention my friends who have made my time and process to obtain my PhD simply easier and more enjoyable. The list of friends is large and it is continuously evolving. For this reason I prefer not to mention particular names. Nevertheless, not mentioning names does not change the essence of these words since they know without a doubt who I am referring to.

I am in debt with professors, mentors and colleagues throughout my PhD. Firstly, I thank my advisor Prof. Jean-Luc Guermond for giving me the opportunity of working with him, for his patience, help and support. Secondly, I acknowledge the help of my co-advisor Prof. Meinhard T. Schobeiri and committee members Prof. Bojan Popov and Prof. Wolfgang Bangerth. It is important for me to emphasize the gratitude I feel to them not only for being members of my committee but also for their help and time during different stages of my PhD.

I thank the BLAST team at Lawrence Livermore National Laboratory (LLNL) for the extremely productive summer internship I spent at LLNL. In particular to Vladimir Z. Tomov, Tzanio Kolev, Robert N. Rieben and Veselin A. Dobrev. In

addition, I am thankful with Prof. Andrea Bonito for the time invested in several meetings to discuss different aspects of the work and research I was involved during the first stages of my PhD. I am also in debt with Prof. David I. Ketcheson for the opportunity of spending two summer internships and one winter internship at KAUST. Time that I found very productive. I am grateful and acknowledge the collaboration with Travis B. Thompson for some of the results in chapter 5. In addition, I need to mention all the Professors at Texas A&M that spent their time and effort providing and giving the excellent courses I had the opportunity to take during my PhD. Finally, I want to thank my colleagues in this journey to obtain a PhD for providing continuous and interesting talks and discussions.

The work in chapter 4 was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 during a summer internship at LLNL. See LLNL-JRNL-684083. The work in all other chapters is based upon work supported in part by the National Science Foundation grant DMS-1217262, by the Air Force Office of Scientific Research, USAF, under grant/contract number FA99550-12-0358 and by the Army Research Office under grant/contract number W911NF-15-1-0517. In addition, I received a complementary support under the scholarship "Beca Complemento" from the Mexican Secretariat of Public Education (SEP) through the DGRI.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

In fluid mechanics the interaction of fluids with distinguishable material properties (e.g. water and air) is referred to as multiphase flow. This problem is important due to its wide range of applications such as water-oil-gas interaction within reservoirs and oil industry equipment, heat exchangers, combustion problems, water-air interaction as a substage of fluid solid interaction and many others. In this work we consider the problem of two-phase incompressible flow. Moreover, we assume the fluids (also known as phases) don't mix; i.e., at every moment it is possible to distinguish one-phase from the other. We are interested in the numerical solution of this problem.

1.1 Literature review

1.1.1 Fluid motion: model and description

The problem of two-phase incompressible flow is modeled by the Navier-Stokes equations with variable material parameters; i.e., variable density and viscosity. The Navier-Stokes equations describe the motion of fluids. They can be derived from conservation principles (conservation of mass and momentum) and can be seen as Newton's second law for fluids. Any initial state of two-phases determines density and viscosity fields. From these fields one needs to solve the Navier-Stokes equations to obtain a velocity field that determines the infinitesimal motion of the initial phases, which yields new density and viscosity fields.

We consider finite elements to obtain a finite dimensional representation of the equations, which requires the use of a computational domain. There exists two main approaches to describe the motion of fluids: Lagrangian and Eulerian descriptions [59]. With Lagrangian algorithms the nodes of the computational domain move via

the physical velocity field. This implies that every node is related to a single point in the fluid and their location is always the same. These methods are convenient for tracking material interfaces. However, they can't be used for large fluid motion since the computational grid can easily get entangled. In Eulerian methods the fluid movement occurs on a fixed computational grid. These methods can handle large fluid deformations but (since the motion is independent on the computational domain) they require a representation of the interface. An alternative that merges good characteristics of each description is known as Arbitrary Lagrangian-Eulerian (ALE) method [12,37,41]. In ALE methods the nodes are allowed to move arbitrarily. They can move by the fluid velocity (as in Lagrangian algorithms), by some other velocity (e.g., a smooth version of the fluid velocity) or they can be held fixed (as in Eulerian codes). An extensive description of Lagrangian, Eulerian and ALE methods for fluid motion can be found (for instance) in [5]. In this work we use an Eulerian description of the fluid motion.

1.1.2 Interface representation in Eulerian descriptions

As explained before, Eulerian algorithms require a representation of the interface. There is an extensive list of methods to treat material interfaces. Popular choices include the Volume of Fluid (VOF) method [38] and level set techniques [65, 71]. The VOF method uses a characteristic function to define the location of the phases; e.g., define the function to be one for fluid A and zero for fluid B. The average of this function over any given cell defines the fraction of the cell occupied by fluid A. Therefore, cells with average between zero and one contain an interface. The characteristic function is transported via the fluid velocity obtained by solving the Navier-Stokes equations. Finally, an interface reconstruction is required to represent the boundary of the phases. The level set method represents the interface of the

phases using a function called the level set function (e.g. a signed distance function from the interface). An interface is defined by a prescribed value of the level set function (e.g., zero). Therefore, the phases are easily determined by the value of the function. The level set function is transported via the fluid velocity obtained by solving the Navier-Stokes equations. The main disadvantage of the VOF method is the difficulty in a-posteriori interface reconstruction. Level set methods don't require interface reconstruction; however, they suffer from loss of area enclosed by the interface. There are hybrid methods combining ideas of the VOF and the level set method. See for instance [14, 42, 70]. In this work we use the level set method.

1.1.3 Level set method

As previously discussed, the level set represents the interface between phases as a prescribed value of a level set function. Regardless of the level set function one needs to solve the transport equation. It is well known that solving this equation (and hyperbolic partial differential equations in general) tends to introduce non-physical oscillations near large gradients. There is an extensive choice of methods to reduce this behavior. We discuss some of them in the following paragraphs. In general, all of these techniques introduce some type of artificial dissipation to reduce non-physical oscillations. Unfortunately, these methods tend to also dissipate the solution. Dissipation increases the loss of the area enclosed by the level set, which produces loss of area on one of the fluids. For this reason, it is common, within the level set methodology, to introduce a reinitialization stage, which is meant to recover the original shape of the level set function and, hence, reduce loss of area. When the level set function is a signed distance function the reinitialization can be performed following [71, 76] and others. See [6, 15, 17, 28, 65, 69, 76] for some examples of applications of these reinitialization ideas.

Another popular level set function, which we consider in this work, is a smoothed Heaviside function. See for example [10, 61, 63]. In this case one can (for instance) define the function to be negative and positive one for each of the phases. The two states are then connected via a smooth but sharp profile. Again the zero level set represents the interface of the phases. In this case, the reinitialization consists on sharpening the interface. To do this we use an artificial compression operator as in [29, 30]. It is important to do this without introducing non-physical oscillations or other instabilities.

1.1.4 Monotonicity treatment

It is well known that numerical solutions of hyperbolic partial differential equations are prone to produce non-physical oscillatory behavior. There is an extensive list of methodologies to reduce or eliminate this behavior. To do this it is useful to understand some notions and properties of the exact solution. We consider divergence free velocities. Under this setting the transport equation can be written as a linear scalar conservation law. Some important properties of the solution of this conservation law are that the solution is monotone, Total Variation Diminishing (TVD), monotonicity preserving [56], Maximum Principle Preserving (MPP) [79] and others. It is natural to desire the solution of a numerical method to preserve some (or all) of these properties. For example, monotone methods are TVD, monotonicity preserving and MPP [56], always converge [11] and satisfy a discrete entropy condition [11,33]. Popular examples of monotone methods are given by Godunov [19] and Lax-Friedrichs [54]. Requiring a method to be monotone is, however, too restrictive. It is shown in [33] that a monotone method is at most first-order accurate. To obtain better than first-order convergence, it is necessary to impose less restrictive conditions. For example, it is common to develop TVD and MPP methods.

It is known, by Godunov's theorem [19], that a monotonicity preserving linear method is at best first-order accurate. Being monotonicity preserving is a weaker restriction than being monotone and TVD; moreover, TVD methods (and hence monotone methods) are monotonicity preserving [56]. Therefore, linear TVD methods are at best first-order accurate. For this reason, to achieve better than first-order convergence, nonlinear methods have to be used. In this work we concentrate on MPP methods, a weaker restriction. However, we impose this condition locally. The numerical solution of a locally maximum principle preserving method at some point in space is bounded by the solution at the previous time step around some neighborhood.

An obvious attempt to obtain high-order nonlinear methods to achieve solutions with some kind of monotonicity constraint (monotone, TVD, MPP, etc.) is to use consistent high-order nonlinear artificial viscosity terms. This idea starts with the work by [77] and is also suggested in [53]. Many more nonlinear approaches have been proposed. In this work we consider a nonlinear artificial viscosity (based on the entropy residual of the solution) proposed in [25]. Nevertheless, it is difficult to introduce the correct amount of artificial viscosity to guarantee the solution posses the desired monotonicity property. For this reason, it is more common to impose the monotonicity restriction directly. There are different alternatives to do this.

Within finite volume methods, it is common to use slope limiters to impose monotonicity constraints. For instance, a class of methods known as Monotonic Upstream-Centered Scheme for Conservation Laws (MUSCL) [64], and in a series of publications [72–75], extend the ideas of the Godunov scheme to second-order via higher-order reconstructions (instead of the piecewise constant reconstruction in the Godunov's method) and using slope limiters to obtain TVD methods. Based on these ideas, in [13] and later in [57], second-order TVD methods are proposed also

for unstructured triangular grids.

Following this methodology, TVD second-order methods have been developed. When higher than second-order is desired, TVD schemes near local extrema degenerate to first-order in the L^∞ norm and second-order in the L^1 norm [32]. A common approach to overcome this barrier is to relax the monotonicity restriction near local extrema. Popular examples are UNO [34], ENO [32, 35] and WENO [58] methods.

Another popular approach to impose maximum principle preservation is to use flux limiting. Flux limiting methods utilize a low- and a high-order solution selectively. The idea is to use the high-order solution when the solution is smooth and switch to the low-order method near discontinuities. This is usually done via nonlinear limiting factors computed based on the solution. A popular flux limiting method, which we use extensively in this work, is the Flux Corrected Transport (FCT) introduced for a one dimensional linear problem using finite differences in [7]. Later in [78] the methodology is presented in a more generalized format for multi-dimensions and considering nonlinear problems. An extensive and detailed description of this method can be found in [47]. A similar class of methods in [31, 36], known as hybrid self-adjusting schemes, automatically and smoothly switches between a first- and a second-order solution depending on the steepness of the solution. The method uses an "automatic switch" parameter to smoothly and automatically switch between the two methods.

1.2 Objectives

The objective of this work is to develop numerical methods to solve the transport equation reducing or eliminating the non-physical oscillations introduced by the numerical methods. We assume the velocity field is divergence free and consider the transport equation in conservation form. To control the oscillatory behavior we de-

velop methods that satisfy the maximum principle locally. We use continuous and discontinuous Galerkin finite element methods with high-order spaces. Afterwards, we use some of these methods to transport a level set function given by a smoothed Heaviside profile with a reinitialization given by a sharpening operator. Finally, we are interested in using the level set and a Navier-Stokes solver to simulate two-phase incompressible flows. We now specify more clearly the objectives of the main chapters of this work.

1.2.1 High-order Maximum Principle Preserving methods with continuous Galerkin finite elements

Within the context of continuous Galerkin finite elements we start with the work in [24] where first-order spaces are used and second-order MPP solutions are achieved. Our objective is to use high-order spaces to extend these results to higher-order accuracy.

1.2.2 High-order Maximum Principle Preserving methods with discontinuous Galerkin finite elements

Using discontinuous Galerkin finite elements we start with the work in [2] where third order MPP solutions are achieved. In this work they focus on the linear problem and apply it to remap a solution between two meshes coming from an Arbitrary Lagrangian Eulerian (ALE) simulation. Applying the same technique to higher order spaces (around 4th order and above) yields highly oscillatory solutions. It is our aim to propose methods to eliminate these non-physical oscillations.

1.2.3 High-order Maximum Principle Preserving methods with artificial compression for the transport equation

We consider an artificial compression operator based on [29, 30]. Our objective is to incorporate this operator within a stabilized linear scalar conservation law to transport a smoothed Heaviside level set. The artificial compression is meant to reinitialize the level set profile by reducing numerical dissipation. Finally, we intend to use this methodology within the FCT method to obtain MPP solutions.

1.2.4 Multiphase flow

Finally, we intend to use the methods for solving the level set with reinitialization with an incompressible Navier-Stokes solver to simulate multiphase simulations in two and three dimensions.

1.3 Methodology

In general we consider the Flux Corrected Transport (FCT) method as in [7] and [78] to obtain Maximum Principle Preserving (MPP) solutions considering high-order spaces. In addition, we present a new FCT-*like* method to solve the transport equation using discontinuous Galerkin finite elements. Finally, we use artificial compression operators within the context of the FCT method to reduce numerical dissipation. We present now some details on the methodology followed to fulfill each of the objectives.

1.3.1 High-order Maximum Principle Preserving methods with continuous Galerkin finite elements

We start by considering the low-order MPP method in [26] and a high-order continuous Galerkin method. Considering these two solutions we apply the FCT method

on high-order spaces to obtain high-order solutions. See §3.3 for details and results on this method. Using this method on high-order spaces introduces small oscillations, which is true nevertheless the solution is maximum principle preserving. To reduce these oscillations we compute the bounds using a localized stencil that mimics the stencil of first-order spaces. To improve the accuracy we use the iterated FCT method as in [68].

1.3.2 High-order Maximum Principle Preserving methods with discontinuous Galerkin finite elements

Here we start with the method in [2] where the authors use the FCT method with a low-order solution as in [51,52] and a high-order method via a discontinuous Galerkin discretization with upwind Godunov numerical flux [55,67]. The authors use second-order spaces and obtain the expected third-order convergence rates. When high-order spaces (4th-5th and above) are used, non-physical oscillations appear. We propose two methods to solve this problem.

For the first method we modify the definition of the bounds to consider a first-order stencil. By doing this the oscillatory behavior is highly reduced, but not completely eliminated. We also observe lesser quality solutions as higher-order polynomials are considered (for fixed number of degrees of freedom). This is a consequence of having a low-order solution of lesser quality. See §4.4 for details and results on this method.

The second method is a new methodology using discontinuous Galerkin finite elements. We start with two MPP solutions. One is mass conservative and low-order and the other is non mass conservative and high-order. We interpolate from the low- to the high-order solution guaranteeing the solution remains on bounds and recovering mass conservation. See section §4.5 for details and results on this method.

1.3.3 High-order Maximum Principle Preserving methods with artificial compression for the transport equation

We consider an artificial compression operator as in [29, 30]. This operator has the structure of an anti-diffusion operator. Therefore, we can use it within the FCT methodology. We do that using different low-order methods and two approaches. In chapter 5 we describe this methodology in detail and show some results.

1.3.4 Multiphase flow

We are interested in solving multiphase incompressible flow problems. To do this we require a Navier-Stokes solver. In chapter 6 we use a projection scheme based on [27] and use a continuous Galerkin discretization in space. After revisiting the method we validate it via convergence tests. Finally, we present results of flow around low pressure turbine blades at relatively high Reynolds numbers. Finally, in chapter 7 we use the Navier-Stokes solver and the level-set method in §5.3.6.1 to simulate multiphase incompressible flow in two and three dimensions.

1.3.5 Implementation of the numerical methods

We use the MFEM library [60] for all numerical simulations in chapter 4. All numerical experiments in chapters 3, 5, 6 and 7 are performed using deal.II [4]. The visualization is performed via GLVis [18], Paraview [1] and MATLAB R2015a.

The rest of the dissertation is divided as follows. In chapter 2 we review theory and definitions required to understand and develop the work during the subsequent chapters. In chapters 3 and 4 we propose different methodologies to obtain maximum principle preserving solutions using continuous and discontinuous Galerkin finite elements respectively. In chapter 5 we propose two maximum principle preserving

methods to transport and reinitialize a smoothed Heaviside level set using continuous Galerkin finite elements. In chapter 6 we revisit an incompressible Navier-Stokes solver for variable density. Finally, in chapter 7 we use one of the proposed level set methods with the Navier-Stokes solver to simulate two-phase incompressible flow.

2. PRELIMINARIES

This chapter is dedicated to review important and relevant material necessary through the rest of this work. In particular, we present the transport equation, describe the process to obtain the initial condition, review the concept of discrete maximum principle and revisit the Flux Corrected Transport (FCT) and the iterative FCT methods.

2.1 Transport equation

Let $\Omega \subset \mathbb{R}^d$ be an open domain and $0 < T \in \mathbb{R}$ be the final time. The boundary of this domain is denoted as $\partial\Omega$. We consider the transport equation given by

$$\partial_t u(\mathbf{x},t) + \nabla \cdot (\mathbf{v}(\mathbf{x},t)u(\mathbf{x},t)) = 0, \qquad \forall (\mathbf{x},t) \in \Omega \times (0,T), \tag{2.1a}$$

$$u(\mathbf{x},t=0) = u_0(\mathbf{x}), \qquad \forall (\mathbf{x}) \in \Omega, \tag{2.1b}$$

$$u(\mathbf{x},t) = u_b(\mathbf{x},t), \qquad \forall (\mathbf{x},t) \in \partial\Omega^- \times (0,T), \tag{2.1c}$$

where $u : \Omega \times (0,T) \to \mathbb{R}$ is the transported solution, $\mathbf{v} : \Omega \times (0,T) \to \mathbb{R}^d$ is the velocity field with $d = \{1,2,3\}$ being the spatial dimension, $u_0 : \Omega \to \mathbb{R}$ is the initial condition, $u_b : \Omega \times (0,T) \to \mathbb{R}$ is the boundary condition and $\partial\Omega^- = \{\mathbf{x} \in \partial\Omega \mid \mathbf{v} \cdot \mathbf{n} < 0\}$, where $\mathbf{n}$ is the outer normal vector. We assume $\nabla \cdot \mathbf{v} = 0$; i.e., the velocity is divergence free.

2.2 The initial condition

Given a finite element space X_h we consider a basis, say $\mathcal{B} = \{\phi_1, \ldots, \phi_N\}$ where N is the dimension of X_h. Let $u_h \in X_h$ be the finite element approximation of u. We can represent u_h using the basis $\mathcal{B}$; i.e., $u_h(\mathbf{x},t^n) = \sum_i U_i^n \phi_i(\mathbf{x})$ where U_i^n's are

the degrees of freedom of the solution. In this work we consider high-order spaces to achieve high-order accuracy. Therefore, each element in the basis function is a high-order polynomial. For reasons explained in §3.2 and §4.2 we consider positive basis functions given by Bernstein polynomials. These basis functions are not nodal (except for first-order spaces). As a result, to set the initial condition U^0 we can't simply evaluate $u_h(\mathbf{x}, t = 0)$ at the nodal points.

We consider two alternatives. The first approach is to perform a projection onto the finite element space. This can be done by solving

$$\sum_j U_j^0 \int_\Omega \phi_i(\mathbf{x})\phi_j(\mathbf{x})d\mathbf{x} = \int_\Omega u_h(\mathbf{x}, t = 0)\phi_i(\mathbf{x})d\mathbf{x},$$

which requires inverting the mass matrix. This process recovers the high-order accuracy of the space but introduces spurious oscillations. If the solution is smooth these oscillations are small and can be neglected (for some applications). However, if the solution is discontinuous the oscillatory behavior is enlarged and this process is unacceptable. We obtain the initial condition U^0 via a projection in all convergence tests whenever we expect to achieve high-order accuracy.

Another alternative is to use Bernstein polynomials to approximate the initial condition. See chapter 7 of [66] for details and properties of this approximation. Given a function $f(x), \forall x \in [0, 1]$ the Bernstein approximation is given by

$$B_k(f; x) = \sum_{r=0}^{k} f\left(\frac{r}{k}\right) b_{r,k}(x),$$

where $b_{r,k}(x)$ is the r-th Bernstein polynomial of order k. This approximation is monotone which implies that $m \leq f(x) \leq M, \forall x \in [0, 1] \implies m \leq B_k(f; x) \leq M, \forall x \in [0, 1]$. The approximation is at best second-order in the L^1 norm for poly-

nomials of any order. If we associate a grid to the points $x_r = \frac{r}{k}$. Then the control points $f\left(\frac{r}{k}\right)$ are the nodal values of the function f at the nodes of the grid. Considering this and using tensor products to consider more than one spatial dimension (in quadrilateral finite elements) we obtain the approximation of the initial condition to be

$$u_h(\mathbf{x}, t = 0) = \sum_i U_i^0 \phi_i(\mathbf{x}),$$

where U_i^0 is the i-th degree of freedom of the finite element approximation and it is given by the nodal value of the function $u_h(\mathbf{x}, t = 0)$ at the location of the i-th node.

2.3 Discrete Maximum Principle and solution on bounds

It is our aim to obtain methods that preserve the maximum principle locally. Given a solution U_i^n at some time $t = t_n$, the solution U_i^{n+1} satisfies the discrete maximum principle locally if

$$\min_{j \in N_i} U_j^n =: U_i^{\min} \leq U_i^{n+1} \leq U_i^{\max} := \max_{j \in N_i} U_j^n, \tag{2.2}$$

where N_i defines some neighborhood of the i-th degree of freedom. Conventionally, this is given by the sparsity pattern of the transport operator. With continuous Galerkin finite elements it is given by the support of the i-th shape function. With discontinuous Galerkin finite elements it is given by all degrees of freedom on the given cell and on adjacent cells sharing a face with it. It is common to refer to a solution that preserves the maximum principle as being *"on bounds"*. We follow this convention.

2.4 Flux Corrected Transport

In this section we revisit the Flux Corrected Transport (FCT) methodology by [7] and [78]. We also refer to [47] for more details and to [24] and [2] for examples of the FCT method using continuous and discontinuous Galerkin finite elements respectively. The FCT method considers a low-order Maximum Principle Preserving (MPP) method and a high-order non-MPP method and interpolates from the low- to the high-order solution assuring the result is on bounds.

2.4.1 Low-order method

Consider a general low-order method

$$M^L \left(\frac{U^L - U^n}{\Delta t} \right) + TU^n + D^L U^n = 0, \tag{2.3}$$

and assume the solution is on bounds. Here M^L is the lumped mass matrix, T is the transport operator and D^L is a linear diffusive operator that introduces enough dissipation to assure the solution is on bounds. There are some alternatives for the low-order method. In particular, we concentrate on three low-order methods proposed in [23], [26] and [51, 52]. We explain the details of these methods in §3.2.1, §3.2.2 and §4.2 respectively.

Remark 2.4.1.1 (The matrix $-D^L$ is a diffusive operator)**.** *The matrix $-D^L$ is symmetric, has non-positive off-diagonal entries and has zero row and column sum; i.e., $\sum_i D^L_{ij} = 0$ and $\sum_j D^L_{ij} = 0$. These are the typical characteristics of the discretization of the Laplace operator $-\Delta$ [51].*

Remark 2.4.1.2 (Mass conservation)**.** *The method* (2.3) *is assumed to be mass*

conservative in the following sense:

$$\int_\Omega u_h^L(\mathbf{x},t)d\mathbf{x} = \int_\Omega u_h^L(\mathbf{x},t=0)d\mathbf{x} \iff \sum_i U_i^L m_i = \sum_i U_i^0 m_i,$$

where $m_i = \int_\Omega \phi_i(\mathbf{x})d\mathbf{x}$ *is the i-th diagonal element of* M^L.

Remark 2.4.1.3 (First-order accurate)**.** *Since the method* (2.3) *is MPP and linear with respect to the solution* U^n*, by Godunov's theorem [19], it is at most first-order accurate.*

2.4.2 High-order method

Consider a general high-order method given by

$$M\left(\frac{U^H - U^n}{\Delta t}\right) + TU^n + D^H U^n = 0, \tag{2.4}$$

where M is the consistent mass matrix and D^H is a high-order diffusive operator. There are also different alternatives for the high-order method. With continuous Galerkin spaces we use D^H given by an artificial viscosity based on the entropy residual of the solution as presented in [25]. With discontinuous Galerkin finite elements we don't introduce any artificial viscosity since the solution is stabilized via the numerical flux, which is included in the definition of the transport matrix.

2.4.3 Flux limiting

The high-order method (2.4) can be rewritten as

$$M^L(U^H - U^L) = (M^L - M)(U^H - U^n) + \Delta t(D^L - D^H)U^n \tag{2.5}$$

where U^L is the low-order solution given by (2.3) and the right hand side is a flux correction. Note that for any $i = 1, \ldots, N$, $\sum_j (M^L - M)_{ij} = 0$ and $\sum_j (D^L - D^H)_{ij} =$

0, by the properties of the diffusive operators D^L and D^H. Therefore,

$$\begin{aligned}[(D^L - D^H)U^n]_i &= \sum_j (D^L - D^H)_{ij} U_j^n = \sum_{j \neq i} (D^L - D^H)_{ij} U_j^n + (D^L - D^H)_{ii} U_i^n \\ &= \sum_{j \neq i} (U_j^n - U_i^n)(D^L - D^H)_{ij} = \sum_j (U_j^n - U_i^n)(D^L - D^H)_{ij},\end{aligned}$$

and since $D^L - D^H$ is symmetric, the matrix with entries $(U_j^n - U_i^n)(D^L - D^H)_{ij}$ forms a skew-symmetric matrix. Similarly, $\sum_j (M^L - M)_{ij}(U_j^H - U_j^n) = \sum_j (M^L - M)_{ij}(\delta U_j - \delta U_i)$, where $\delta U := U^H - U^n$. Here the matrix with entries $(M^L - M)_{ij}(\delta U_j - \delta U_i)$ also forms a skew-symmetric matrix. Let us introduce the so-called flux correction matrix F with entries

$$f_{ij} := (M^L - M)_{ij}(\delta U_j - \delta U_i) + \Delta t (D^L - D^H)_{ij}(U_j^n - U_i^n). \tag{2.6}$$

The above arguments show that $f_{ij} = -f_{ji}$, i.e., F is skew-symmetric. Then the high-order method (2.4) can be rewritten as

$$U_i^H = U_i^L + m_i^{-1} \sum_j f_{ij}. \tag{2.7}$$

From here we can see that the flux correction improves the accuracy of the low-order method to make it high-order. In addition, it is responsible for the high-order solution to be off bounds. The idea behind FCT is to limit this correction whenever it makes the solution to be off bounds. Following [78] we introduce a symmetric flux limiter matrix α, with entries α_{ij}, and compute the high-order update as follows:

$$U_i^{n+1} = U_i^L + m_i^{-1} \sum_j \alpha_{ij} f_{ij}, \tag{2.8}$$

where the entries of the flux limiter matrix are given by

$$\alpha_{ij} := \begin{cases} \min\left(R_i^+, R_j^-\right) & \text{if } f_{ij} \geq 0, \\ \min\left(R_i^-, R_j^+\right) & \text{if } f_{ij} < 0 \end{cases}, \tag{2.9a}$$

where

$$R_i^+ := \begin{cases} \min\left(1, \frac{Q_i^+}{P_i^+}\right), & \text{if } P_i^+ \neq 0, \\ 1, & \text{if } P_i^+ = 0 \end{cases}, \quad R_i^- := \begin{cases} \min\left(1, \frac{Q_i^-}{P_i^-}\right), & \text{if } P_i^- \neq 0, \\ 1, & \text{if } P_i^- = 0 \end{cases} \tag{2.9b}$$

$$P_i^+ := \sum_j \max(0, f_{ij}), \qquad P_i^- := \sum_j \min(0, f_{ij}), \tag{2.9c}$$

$$Q_i^+ := m_i(U_i^{\max} - U_i^L), \qquad Q_i^- := m_i(U_i^{\min} - U_i^L). \tag{2.9d}$$

Theorem 2.4.3.1 (Maximum Principle)**.** *Assume that U^L satisfies the local discrete maximum principle; i.e., $U_i^{\min} \leq U_i^L \leq U_i^{\max}$ for all $i = 1, \ldots, N$. Then the solution of (2.8) satisfies the local discrete maximum principle; i.e., $U_i^{\min} \leq U_i^{n+1} \leq U_i^{\max}$ for all $i = 1, \ldots, N$.*

Proof. To see this we follow [47, p. 182] and [24]. Assume that $P_i^+ \neq 0$. Using (2.9) we get

$$\begin{aligned} m_i(U_i^{n+1} - U_i^L) = \sum_j \alpha_{ij} f_{ij} &\leq \sum_{j, f_{ij} \geq 0} \alpha_{ij} f_{ij} = \sum_{j, f_{ij} \geq 0} \min(R_i^+, R_j^-) f_{ij} \leq \sum_{j, f_{ij} \geq 0} R_i^+ f_{ij} \\ &\leq \frac{Q_i^+}{P_i^+} \sum_{j, f_{ij} \geq 0} f_{ij} = \frac{Q_i^+}{P_i^+} \sum_j \max(0, f_{ij}) = Q_i^+ = m_i(U_i^{\max} - U_i^L); \end{aligned}$$

therefore, $U_i^{n+1} \leq U_i^{\max}$. If $P_i^+ = 0$, then

$$m_i(U_i^{n+1} - U_i^L) \leq \sum_{j, f_{ij} \geq 0} R_i^+ f_{ij} = \sum_{j, f_{ij} \geq 0} f_{ij} = P_i^+ = 0,$$

and, provided $U_i^L \leq U_i^{\max}$, we get $P_i^+ = 0 \leq m_i(U_i^{\max} - U_i^L)$, which implies $U_i^{n+1} \leq U_i^{\max}$. The lower bound $U_i^{\min} \leq U_i^{n+1}$ is proven similarly. □

Remark 2.4.3.1 (Mass conservation)**.** *The method* (2.8) *is mass conservative in the following sense:*

$$\int_\Omega u_h(\mathbf{x}, t) d\mathbf{x} = \int_\Omega u_h(\mathbf{x}, t = 0) d\mathbf{x} \iff \sum_i U_i^{n+1} m_i = \sum_i U_i^0 m_i,$$

where $m_i = \int_\Omega \phi_i(\mathbf{x}) d\mathbf{x}$*. To see this consider the symmetry properties of* α_{ij} *and* f_{ij} *and get the row sum of* (2.8)

$$\sum_i m_i(U_i^{n+1} - U_i^L) = \sum_i \sum_j \alpha_{ij} f_{ij} = \sum_{i,j} \alpha_{ij} f_{ij} + \alpha_{ji} f_{ji} = \sum_{i,j} \alpha_{ij}(f_{ij} - f_{ij}) = 0.$$

□

2.5 Iterative Flux Corrected Transport

In this section we revisit the iterative Flux Corrected Transport method in [48,49,68]. We recall the FCT method revisited in the previous section:

$$m_i(U_i^{n+1} - U_i^L) = \sum_j \alpha_{ij} f_{ij} \tag{2.10}$$

where U^L is a MPP low-order solution, f_{ij}'s are flux corrections and α_{ij}'s are the flux limiters that prevent the solution to be off bounds. The idea of the iterated FCT is to consider the solution U^{n+1} to be a "low-order" solution on bounds and repeat the

FCT process.

Renaming the solution U^{n+1} in equation (2.10) to be $\tilde{U}^L$ yields

$$m_i(\tilde{U}_i^L - U_i^L) = \sum_j \alpha_{ij} f_{ij}. \tag{2.11}$$

Subtract U^H from equation (2.7) (the high-order method) to obtain

$$m_i(U_i^H - \tilde{U}_i^L) = \sum_j (1 - \alpha_{ij}) f_{ij}, \tag{2.12}$$

where the right hand side is a flux correction that modifies the solution $\tilde{U}^L$ to become the high-order solution U^H. We now apply the flux limitation as explained in the previous section to obtain

$$m_i(U_i^{n+1} - \tilde{U}_i^L) = \sum_j \tilde{\alpha}_{ij}(1 - \alpha_{ij}) f_{ij}, \tag{2.13}$$

where $\tilde{\alpha}_{ij}$ is the flux limiter of the flux $(1-\alpha_{ij})f_{ij}$. We can plug equation (2.11) into (2.13) to obtain

$$m_i(U_i^{n+1} - U_i^L) = \sum_j \alpha_{ij} f_{ij} + (1 - \alpha_{ij})\tilde{\alpha}_{ij} f_{ij}. \tag{2.14}$$

Since the limiters $0 \leq \alpha_{ij} \leq 1$ (see (2.9)) we can interpret the right hand side of (2.14) as a convex combination of two fluxes: a non-limited flux f_{ij} and a limited flux $\tilde{\alpha}_{ij} f_{ij}$. This process can be repeated as many times as desired.

To verify the result of different iterations we use a method proposed in chapter 3. The method, which is presented in §3.4, uses continuous Galerkin finite elements and is based on the FCT methodology as revisited in §2.4. The main idea of this method

is that the bounds are computed using a smaller stencil mimicking a first-order space. In §3.4 we explain all the details of this methodology. Consider an initial condition given by

$$u_h(\mathbf{x}, t = 0) = \begin{cases} 1, & \forall x \in (0.4, 0.6) \\ 0, & \text{otherwise} \end{cases}. \tag{2.15}$$

The domain is given by $\Omega = (0, 1) \subset \mathbb{R}$ and the velocity by $\mathbf{v} = 1$. We consider $\mathbb{Q}_1$, $\mathbb{Q}_2$, $\mathbb{Q}_4$ and $\mathbb{Q}_8$ spaces and adjust the number of cells to have 256 degrees of freedom in all cases. Table 2.1 shows the L^1 error at time $T = 1$ considering different number of FCT iterations. We also show the error normalized with respect to the error by doing a single FCT iteration. An important remark is that doing multiple FCT iterations seems to have a little effect when $\mathbb{Q}_1$ spaces are used. For higher order spaces we obtain a substantial reduction of the error of around 25% by doing two FCT iterations. However, doing more iterations seems to have little effect on the error. Unless otherwise noted we use one FCT iteration for $\mathbb{Q}_1$ spaces and two iterations otherwise.

Iter.	L^1 error	Norm. error
1	1.40E-02	1
2	1.39E-02	0.993
3	1.39E-02	0.991
4	1.39E-02	0.991

(a) $\mathbb{Q}_1$ space

Iter.	L^1 error	Norm. error
1	1.38E-02	1
2	1.05E-02	0.761
3	1.04E-02	0.751
4	1.04E-02	0.750

(b) $\mathbb{Q}_2$ space

Iter.	L^1 error	Norm. error
1	4.36E-02	1
2	3.30E-02	0.757
3	3.29E-02	0.754
4	3.28E-02	0.753

(c) $\mathbb{Q}_4$ space

Iter.	L^1 error	Norm. error
1	4.90E-02	1
2	3.89E-02	0.794
3	3.68E-02	0.750
4	3.08E-02	0.628

(d) $\mathbb{Q}_8$ space

Table 2.1: L^1 error for iterative Flux Corrected Transport. We use $\mathbb{Q}_1$, $\mathbb{Q}_2$, $\mathbb{Q}_4$ and $\mathbb{Q}_8$ spaces and consider different number of FCT iterations.

3. FLUX CORRECTED TRANSPORT WITH CONTINUOUS GALERKIN FINITE ELEMENTS

We start this chapter following the work in [24] where the authors use the Flux Corrected Transport (FCT) methodology with $\mathbb{P}_1$ continuous Galerkin finite elements to obtain a second-order Maximum Principle Preserving (MPP) method for the first-order conservation law. The low-order MPP method is given by [23] and the high-order method is stabilized via a nonlinear artificial viscosity based on the entropy residual of the solution as presented in [25]. The goal of this chapter is to extend these results to third and higher-order. The chapter is split as follows. In §3.1 and §3.2 we present the high- and low-order methods we consider. Afterwards, in §3.3, we apply the FCT method as revisited in §2.4. In §3.4 we use tighter bounds within the FCT methodology. Finally, in §3.5 we present convergence tests and numerical experiments.

3.1 High-order non-Maximum-Principle Preserving method

3.1.1 Spatial discretization

Consider a computational mesh $\mathcal{T}_h$ and define the continuous finite dimensional space $X_h = \{\phi : \phi|_K \in \mathbb{Q}|_K, \forall K \in \mathcal{T}_h, [[\phi]] = 0\}$ where $\mathbb{Q}|_K$ is a polynomial space over the element K. We consider a Galerkin approximation; i.e., we use the space X_h for the trial and test functions. Let $\phi \in X_h$ be a shape function, multiply the transport equation (2.1) by it and integrate over the domain Ω. In addition, let $u_h \in X_h$ be

the finite element approximation of u. The problem becomes find $u_h \in X_h$ such that

$$\int_\Omega (\partial_t u_h)\phi d\mathbf{x} + \int_\Omega \left[\nabla \cdot (\mathbf{v} u_h)\right] \phi d\mathbf{x} = 0. \tag{3.1}$$

This equation can be recast into matrix-vector form as

$$M\frac{\partial U(t)}{\partial t} + T(U(t)) = 0, \tag{3.2a}$$

where $U(t)$ are the degrees of freedom changing in time, M is the consistent mass matrix with entries

$$M_{ij} = \int_\Omega \phi_i \phi_j d\mathbf{x}, \tag{3.2b}$$

and $T(U(t))$ is a discretization of the transport operator acting on the solution U^n. We use different expressions depending on the method. The details are given in the corresponding sections (see (3.4b) and (3.6b)).

In this chapter we don't consider any stabilization for the high-order method. In chapter 5 we incorporate a nonlinear high-order stabilization to (3.2).

3.1.2 Time discretization

For simplicity we present the full discretization considering Forward Euler integration in time. However, we extend the results to high-order approximations in time via Strong Stability Preserving (SSP) methods [20]. Indeed, all numerical experiments, unless otherwise noted, are performed via a third-order (with three stages) Runge-Kutta SSP method. The time discretization of (3.2) via Forward Euler is given

by

$$M\left(\frac{U^H - U^n}{\Delta t}\right) + T(U^n) = 0, \tag{3.3}$$

where U^H and U^n are the degrees of freedom at time t^{n+1} and t^n respectively.

3.2 Low-order Maximum Principle Preserving method

We consider two different linear methods that preserve the maximum principle. These methods are given by [23] and [26]. Since we attempt to obtain high-order convergence rates (better than 2) we must consider approximations on high-order polynomial spaces (at least $\mathbb{Q}_2$). Under the standard FCT theory for finite elements the low- and high-order methods are approximated by the same finite element space. This means that we need to obtain the low-order solution on polynomial spaces of any order. For this reason we explore in this section the behavior of the low-order methods by [23] and [26] for finite element spaces of different order.

3.2.1 Graph Laplacian based low-order method

We study first the behavior of the low-order method in [23]. This method is given by

$$M^L\left(\frac{U^L - U^n}{\Delta t}\right) + T(U)^n + D^L U^n = 0, \tag{3.4a}$$

where M^L is the diagonal lumped mass matrix whose entries are given by $M^L_{ij} = \int_\Omega \phi_i \delta_{ij} d\mathbf{x}$ with δ_{ij} being the Kronecker delta, $T(U)$ is the column vector with entries given by

$$T(U(t))_i = \int_\Omega \nabla \cdot (\mathbf{v} u_h)\phi_i d\mathbf{x}. \tag{3.4b}$$

and D^L is a dissipative matrix whose entries are given by

$$D_{ij}^L = \sum_{K \in \mathcal{T}_h} \nu_K^L b_K(\phi_i, \phi_j), \tag{3.4c}$$

with

$$b_K(\phi_i, \phi_j) = \begin{cases} -\frac{|K|}{n_K - 1}, & \text{if } i \neq j, i, j \in \mathcal{I}_K \\ |K|, & \text{if } i = j, i, j \in \mathcal{I}_K \\ 0, & \text{otherwise} \end{cases}, \tag{3.4d}$$

$$\nu_K^L = \max_{\substack{i,j \in \mathcal{I}_K \\ i \neq j}} \frac{\left| \int_{S_{ij}} \nabla \cdot (\mathbf{v}\phi_j)\phi_i d\mathbf{x} \right|}{-\sum_{T \subset S_{ij}} b_T(\phi_i, \phi_j)}, \tag{3.4e}$$

where $K \in \mathcal{T}_h$ is a cell, n_K is the number of degrees of freedom in K, $\mathcal{I}_K$ is the index set of all degrees of freedom on cell K and $S_{ij} = S_i \cap S_j$ with S_i being the support of the i-th shape function and similarly for S_j.

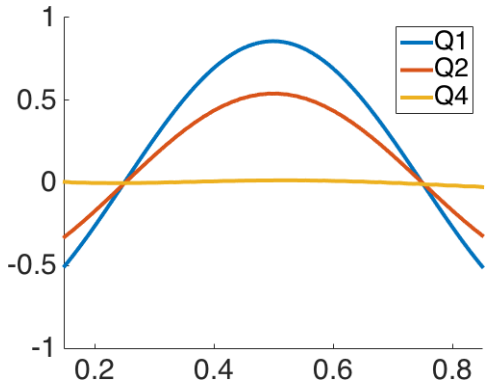
Remark 3.2.1.1 (Properties of the low-order method (3.4))**.** *The method* (3.4) *is maximum principle preserving under a CFL condition, the matrix D^L is a graph Laplacian dissipative matrix and the method is mass conservative (assuming no inflow/outflow of the domain). For the details and proofs of these remarks see [23].*

We now study the behavior of this method under polynomial spaces of different order and with different basis functions. We consider an initial profile given by

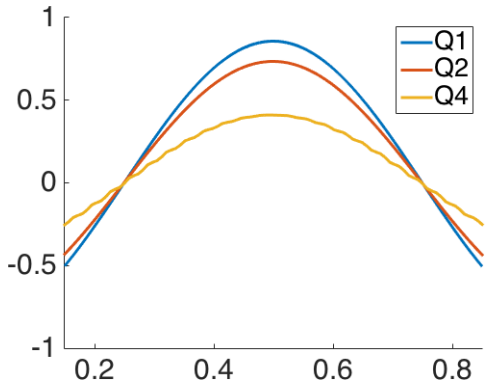
$$u_h(\mathbf{x}, t = 0) = \cos\left(2\pi(x - 0.5)\right), \tag{3.5}$$

over $\Omega = (0, 1) \subset \mathbb{R}$ and velocity $\mathbf{v} = 1$. We impose periodic boundary conditions and compute the solution at time $T = 1$. In figure 3.1a we show the results using

$\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_4$ spaces with nodal basis functions given via Lagrange polynomials. The number of cells is adjusted to have 256 degrees of freedom in all cases. We try the same experiment using positive modal basis functions given by Bernstein polynomials. The results are shown in figure 3.1b. From this experiment we observe that more dissipation is introduced as the order of the space is increased. In addition, non-physical oscillations are introduced in the solution. For these reasons we avoid using this low-order method on any space different than $\mathbb{Q}_1$; consequently, we can't use this low-order method within a standard FCT methodology to obtain better than second-order convergence.



(a) Nodal basis via Lagrange polynomials



(b) Modal basis via Bernstein polynomials

Figure 3.1: Low-order (graph Laplacian based) method with positive v.s. nodal basis functions using continuous Gelerkin finite elements. We consider the low-order method (3.4) and use $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_4$ spaces. The number of cells is adjusted to have 256 degrees of freedom in all situations.

3.2.2 Edge based low-order method

We now study the behavior of the low-order method in [26]. The method is given by

$$M^L \left(\frac{U^L - U^n}{\Delta t} \right) + T(U^n) + D^L U^n = 0, \tag{3.6a}$$

where M^L is the diagonal lumped mass matrix (defined in the previous section), $T(U^n)$ is the column vector with entries

$$T(U^n)_i = \sum_j (\mathbf{v} u_h)_j \cdot \mathbf{c}_{ij} \tag{3.6b}$$

and D^L is a dissipative matrix whose elements are given by

$$D^L_{ij} = -\max(|\mathbf{v}_i \cdot c_{ij}|, |\mathbf{v}_j \cdot c_{ji}|), \quad \forall i \neq j, \tag{3.6c}$$

and $D^L_{ii} = -\sum_{j \neq i} D^L_{ij}$. Here

$$\mathbf{c}_{ij} = \int_\Omega \phi_i \nabla \phi_j d\mathbf{x}, \tag{3.6d}$$

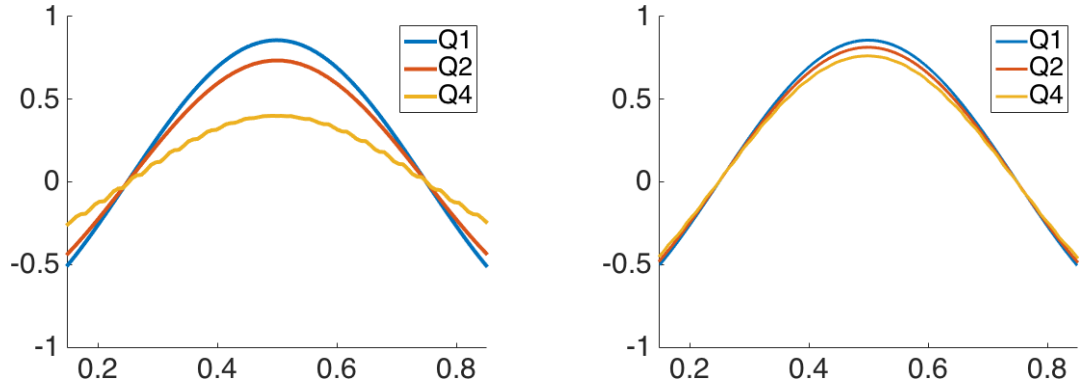
and $(\mathbf{v} u_h)_j, j = 1, ..., N$ are the degrees of freedom of the projection of $\mathbf{v} u_h$ onto the finite element space; i.e., they are given by

$$\sum_j (\mathbf{v} u_h)_j \int_\Omega \phi_i \phi_j d\mathbf{x} = \int_\Omega (\mathbf{v} u_h) \phi_i d\mathbf{x}. \tag{3.6e}$$

Remark 3.2.2.1 (Properties of the low-order method (3.6))**.** *The method* (3.6) *is maximum principle preserving under a CFL condition, the matrix D^L is an edge based dissipative matrix and the method is mass conservative (assuming no inflow/outflow*

of the domain). For the details and proofs of these remarks see [26].

We now study the behavior of this method using different polynomial spaces with nodal and modal basis functions. We consider the same problem as in the previous section and show the results in figure 3.2 using $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_4$ spaces, the number of cells is adjusted to have 256 degrees of freedom in all cases. With nodal basis we observe an slightly oscillatory behavior as the order of the polynomial space is increased. This doesn't happen with the positive modal basis via Bernstein polynomials.



(a) Nodal basis via Lagrange polynomials

(b) Modal basis via Bernstein polynomials

Figure 3.2: Low-order (edge based) method with positive v.s. nodal basis functions using continuous Gelerkin finite elements. We consider the low-order method (3.6) and use $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_4$ spaces. The number of cells is adjusted to have 256 degrees of freedom in all situations.

Remark 3.2.2.2 (More dissipative solutions with higher-order spaces)**.** *It is clear, from figure 3.2b, that the solution is more dissipated as the order of the polynomial is increased. This is crucial to consider when the FCT methodology is applied. Within*

FCT an interpolation from the low- to the high-order solution is performed. Having a low-order solution more dissipated decreases the accuracy of the FCT solution.

Remark 3.2.2.3 (Similarity with low-order method by [51,52])**.** *For the linear problem, method* (3.6) *introduces a dissipative matrix very similar to that used in [51, 52]. It is not surprising that both low-order methods behave similarly for the linear problem under different spaces and with different basis functions. See section 4.2 for experiments using the low-order method by [51, 52]. It is important, however, to remark that the methods differ for nonlinear problems, as noted in [26].*

From these results we decide to use the low-order method by [26] with modal basis functions via Bernstein polynomials. We expect, however, to lose accuracy on the FCT solution as we use higher-order polynomial spaces.

3.2.2.1 Numerical validation

We perform a convergence test in space for the low-order method (3.6). Consider the same problem as in §3.2.1. We use positive basis functions given by Bernstein polynomials. The results for different spaces are shown in table 3.1. Observe that the error with the 8-th order space is about 3 times smaller than with the 1-st order space; nevertheless, the number of degrees of freedom is vastly (8 times) larger. This indicates that the quality of the low-order solution decreases as we consider higher-order spaces. We expect this to affect the quality of the FCT solution for high-order spaces. This was also observed in §3.2.2.

3.3 Edged-based Flux Corrected Transport with continuous Galerkin finite elements

In this section we consider the standard edge-based FCT methodology, as revisited in §2.4, using the low- and high-order methods in sections §3.2.2 and §3.1 respectively.

Cells	$\mathbb{Q}_1$	rate	$\mathbb{Q}_2$	rate	$\mathbb{Q}_3$	rate	$\mathbb{Q}_4$	rate
32	2.94E-01		2.15E-01		1.75E-01		1.52E-01	
64	1.69E-01	0.796	1.19E-01	0.860	9.46E-02	0.889	8.10E-02	0.906
128	9.11E-02	0.894	6.23E-02	0.928	4.92E-02	0.944	4.19E-02	0.952
256	4.73E-02	0.946	3.19E-02	0.963	2.51E-02	0.971	2.13E-02	0.976
512	2.41E-02	0.972	1.62E-02	0.981	1.27E-02	0.985	1.07E-02	0.988
Cells	$\mathbb{Q}_5$	rate	$\mathbb{Q}_6$	rate	$\mathbb{Q}_7$	rate	$\mathbb{Q}_8$	rate
32	1.36E-01		1.25E-01		1.16E-01		1.09E-01	
64	7.21E-02	0.916	6.58E-02	0.923	6.09E-02	0.929	5.70E-02	0.933
128	3.71E-02	0.957	3.38E-02	0.961	3.12E-02	0.964	2.92E-02	0.966
256	1.88E-02	0.978	1.71E-02	0.980	1.58E-02	0.982	1.48E-02	0.983
512	9.49E-03	0.989	8.61E-03	0.990	7.95E-03	0.991	7.42E-03	0.991

Table 3.1: L^1 convergence of edge based low-order method (3.6).

We consider first 1D simulations with discontinuous data given by

$$u_h(\mathbf{x}, t=0) = \begin{cases} 1, & \forall x \in (0.4, 0.6) \\ 0, & \text{otherwise} \end{cases}. \tag{3.7}$$

The domain is given by $\Omega = (0,1) \subset \mathbb{R}$ and the velocity by $\mathbf{v} = 1$. The initial condition is obtained as explained in §2.2. We use $\mathbb{Q}_1$, $\mathbb{Q}_2$, $\mathbb{Q}_4$ and $\mathbb{Q}_8$ spaces and consider different refinements. For each refinement we adjust the number of cells to have the same number of degrees of freedom between all spaces. The results are shown in figure 3.3. We observe immediately two problems. Oscillations are present for high-order spaces and we obtain more dissipated solutions as we increase the order of the space. We propose to eliminate the oscillations by considering tighter bounds; in particular, we compute bounds mimicking first-order stencils. This is analogous to the method proposed in §4.4 for discontinuous spaces. Since the FCT solution interpolates from the low- to the high-order solution and the low-order solution is clearly more dissipated as the order of the space is increased, see figure 3.2b, we

expect to have more dissipation in the FCT solution as the order of the space is increased. A convergence study is needed to determine the feasibility of the high-order methods to give better results after some refinements. We do this in §3.5.

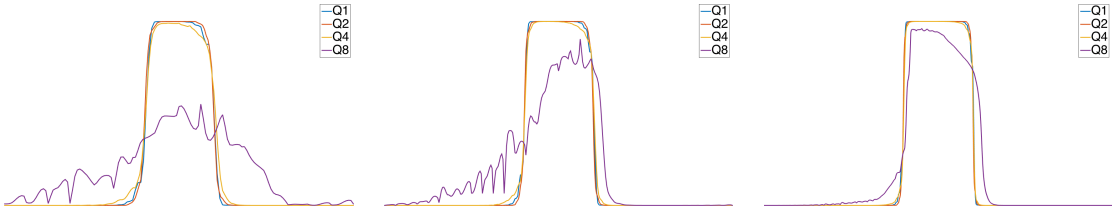


Figure 3.3: Standard Flux Corrected Transport method on a 1D problem with discontinuous initial condition using continuous Galerkin finite elements. The method is given by (2.8) with the low- and high-order methods in sections §3.2.2 and §3.1 respectively. We consider different polynomial spaces and multiple refinements. For each refinement the number of cells is adjusted to have the same number of degrees of freedom in all spaces.

3.4 Localized Flux Corrected Transport with continuous Galerkin finite elements

In this section we propose to localize the bounds as in §4.4. Under the FCT methodology with continuous Galerkin finite elements the bounds (for a given degree of freedom) are computed considering the support of the corresponding shape function. For high-order spaces this process might involve a high number of degrees of freedom. The method loses locality with respect to degrees of freedom. For this reason we consider tighter bounds; in particular, we localize the bounds by mimicking the stencil of a first-order space. Let N_i be the conventional neighborhood for the i-th

degree of freedom. For the $\mathbb{Q}_k$ space, we use the tighter bounds given by the stencil

$$N_i^* = \left\{ j \in N_i : \ \text{dist}(i,j) \leq \frac{\sqrt{d}}{k} \right\}, \tag{3.8}$$

where $d \in \{1,2,3\}$ is the space dimension and $\text{dist}(i,j)$ is the Euclidean distance between the i-th and j-th degrees of freedom's images on the reference element. Defining N_i^* with respect to the reference element makes the approach applicable to unstructured grids. In figure 3.4 we consider representative degree of freedom in thick blue and show the tighter stencil (3.8) in red. The stencil also includes the blue degree of freedom. We show the rest of the conventional stencil with black marks. We consider three possible situations. First when the representative degree of freedom is inside the cell. The second scenario is when the degree of freedom is at the face of two cells. Finally, we consider the situation when a degree of freedom is at the vertex of a cell.

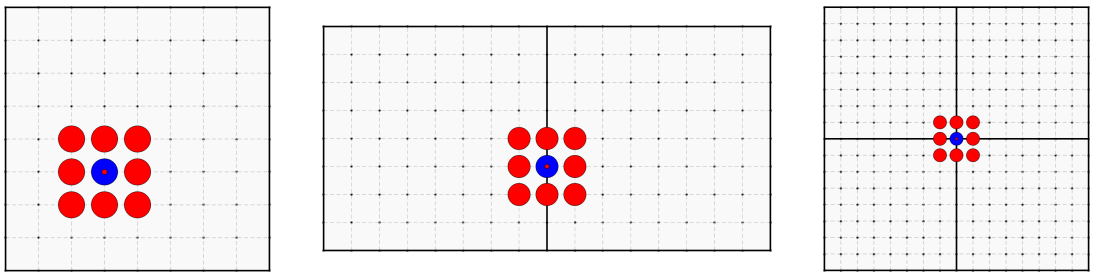


Figure 3.4: Full and localized stencil to compute bounds using continuous spaces. The thick blue marks represent the degree of freedom for which we compute the bounds. The red thick marks represent the degrees of freedom included to compute the bounds. The blue degree of freedom are also considered to compute the bounds. The non-thick black marks are the rest of degrees of freedom on the conventional stencil. We consider three situations. On the left the degree of freedom is inside the cell. On the middle we consider a degree of freedom at the face of two adjacent cells. On the right we consider a degree of freedom at the vertex of four cells.

Remark 3.4.0.4 (Low-order method is non Maximum Principle Preserving on the tighter bounds)**.** *The low-order method* (3.6) *is guaranteed to produce a MPP solution on the conventional bounds. Since the tighter bounds consider a smaller set of degrees of freedom there is no guarantee the low-order solution is MPP in this set.*

Due to remark 3.4.0.4 we can't use the standard FCT with the bounds in (2.2) considering the stencil (3.8). Therefore, we modify the bounds in (2.2) to be

$$U_i^{\min} = \min\left(U_i^L, \min_{j\in N_i^*} U_j^n\right), \tag{3.9a}$$

$$U_i^{\max} = \max\left(U_i^L, \max_{j\in N_i^*} U_j^n\right), \tag{3.9b}$$

which assure the low-order solution is on bounds. We can apply now the FCT method. We repeat the simulations in §3.3 and show the results in figure 3.5. The oscillatory behavior is highly reduced. For all these simulations we obtained better results with $\mathbb{Q}_2$ than $\mathbb{Q}_1$ spaces. However, as we consider higher-order spaces we obtain more dissipated solutions. This is expected since the low-order solution is more dissipative for higher-order spaces, see figure 3.2b. In the next section we perform convergence studies to asses if using higher- (than second)-order spaces can potentially lead to better solutions (for a fixed number of degrees of freedom).

3.5 Numerical experiments

In this section we perform numerical experiments using the edge-based FCT method in §3.3 with the full or conventional stencil N_i and the edge-based FCT method in §3.4 with the localized stencil N_i^* given by (3.8). We perform convergence tests for a two dimensional smooth profile without local extrema, a one dimensional discontinuous profile and a one dimensional smooth profile with local extrema. In addition, we solve two benchmark problems in two dimensions.

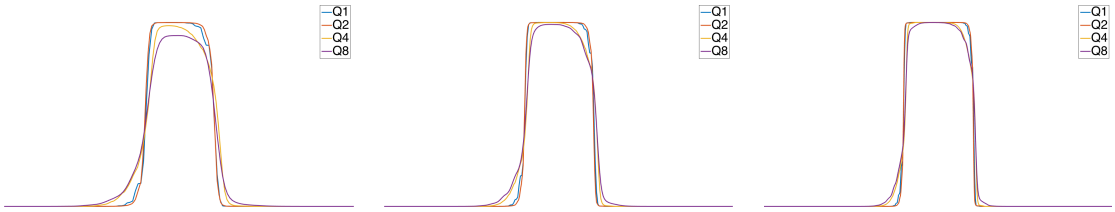


Figure 3.5: Localized Flux Corrected Transport method on a 1D problem with discontinuous initial condition using continuous Galerkin finite elements. We use the low- and high-order methods in sections §3.2.2 and §3.1 respectively. We consider different polynomial spaces and multiple refinements. For each refinement the number of cells is adjusted to have the same number of degrees of freedom in all spaces.

3.5.1 Convergence test: two dimensional smooth profile without local extrema

Consider an initial condition given by

$$u_h(x, y, t = 0) = \tanh((y - 0.5)/0.25), \tag{3.10a}$$

over $\Omega = (0, 1) \times (0, 1)$ with velocity

$$\mathbf{v} = \begin{bmatrix} \sin(\pi x)\cos(\pi y)\sin(2\pi t) \\ -\cos(\pi x)\sin(\pi y)\sin(2\pi t) \end{bmatrix} \tag{3.10b}$$

Since the velocity field is periodic and the problem is linear the exact solution at $T = 1$ coincides with the initial condition. We consider $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_3$ spaces. For this experiment we use a 4-th order with 4 stages Runge Kutta method. Tables 3.2a and 3.2b show the convergence results for the edge-based FCT method with the full stencil N_i and with the localized stencil N_i^* respectively. We obtain the expected (optimal) convergence rates.

Cells	$\mathbb{Q}_1$ space	rate	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_3$ space	rate
64	6.36E-03		6.53E-04		5.18E-05	
128	1.21E-03	2.38	8.97E-05	2.86	1.47E-06	5.14
256	2.83E-04	2.10	1.17E-05	2.94	8.39E-08	4.12
512	6.88E-05	2.03	1.42E-06	3.03	5.00E-09	4.06

(a) Edge-based FCT with the standard or full stencil N_i

Cells	$\mathbb{Q}_1$ space	rate	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_3$ space	rate
64	6.40E-03		6.73E-04		5.21E-05	
128	1.21E-03	2.39	9.14E-05	2.88	1.47E-06	5.14
256	2.83E-04	2.10	1.17E-05	2.96	8.41E-08	4.12
512	6.88E-04	2.03	1.42E-06	3.04	5.01E-09	4.06

(b) Edge-based FCT with the localized stencil N_i^*

Table 3.2: L^1 convergence of maximum principle preserving methods using continuous Galerkin finite elements for s smooth solution that is monotone. We use a 4-th order with 4 stages Runge Kutta method.

3.5.2 Convergence test: one dimensional discontinuous profile

Now consider the same problem with discontinuous initial data as in §3.3 The initial condition is given by

$$u_h(\mathbf{x}, t=0) = \begin{cases} 1, & \forall x \in (0.4, 0.6) \\ 0, & \text{otherwise} \end{cases}. \tag{3.11}$$

The domain is given by $\Omega = (0,1) \subset \mathbb{R}$ and the velocity by $\mathbf{v} = 1$. We consider $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_4$ spaces. Tables 3.3a and 3.3b show the convergence results for the edge-based FCT method with the full stencil N_i and with the localized stencil N_i^* respectively.

DOFs	$\mathbb{Q}_1$ space	rate	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_4$ space	rate
256	1.44E-02		1.13E-02		2.04E-02	
512	9.57E-03	0.58	6.73E-03	0.74	1.02E-02	1.00
1024	6.54E-03	0.55	4.12E-03	0.70	5.21E-03	0.96
2048	3.83E-03	0.77	2.60E-03	0.66	2.63E-03	0.98
4096	2.13E-03	0.84	1.70E-03	0.61	1.33E-03	0.98

(a) Edge-based FCT with full stencil N_i

DOFs	$\mathbb{Q}_1$ space	rate	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_4$ space	rate
256	1.44E-02		1.06E-02		3.35E-02	
512	9.57E-03	0.58	6.27E-03	0.75	2.29E-02	0.55
1024	6.54E-03	0.55	3.86E-03	0.70	1.33E-02	0.78
2048	3.83E-03	0.77	2.44E-03	0.65	6.92E-03	0.94
4096	2.13E-03	0.84	1.59E-03	0.61	3.50E-03	0.98

(b) Edge-based FCT with localized stencil N_i^*

Table 3.3: L^1 convergence of maximum principle preserving methods using continuous Galerkin finite elements for a discontinuous solution.

3.5.3 Convergence test: one dimensional smooth profile with local extrema

Finally we consider as initial condition

$$u_h(x, t = 0) = \cos(2\pi(x - 0.5)),$$

over $\Omega = (0, 1) \subset \mathbb{R}$ with velocity $\mathbf{v} = 1$. We impose periodic boundary conditions and use the initial condition as exact solution at $T = 1$. We use $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_3$ spaces. In tables 3.4a and 3.4b we show the convergence tests via the edge-based FCT with full stencil N_i and with the localized stencil N_i^* respectively. No better than (slightly higher than) second-order is achieved (in the L^1 norm). This is a common problem for methods imposing some type of monotonicity constraint and it is an active area of research. See for instance [34] where it is shown that Total Variation Diminishing

(TVD) methods can't achieve better than second-order convergence (in the L^1 norm) around local extrema. We make more comments on this problem in §4.6.3.

Cells	$\mathbb{Q}_1$ space	rate	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_3$ space	rate
128	7.43E-04		2.99E-04		2.89E-04	
256	1.76E-04	2.07	6.64E-05	2.17	8.05E-05	1.84
512	4.31E-05	2.03	1.42E-05	2.22	2.24E-05	1.84
1024	1.04E-05	2.04	3.03E-06	2.23	4.74E-06	2.24
2048	2.58E-06	2.01	6.26E-07	2.27	9.59E-07	2.30

(a) Edge-based FCT with the standard or full stencil N_i

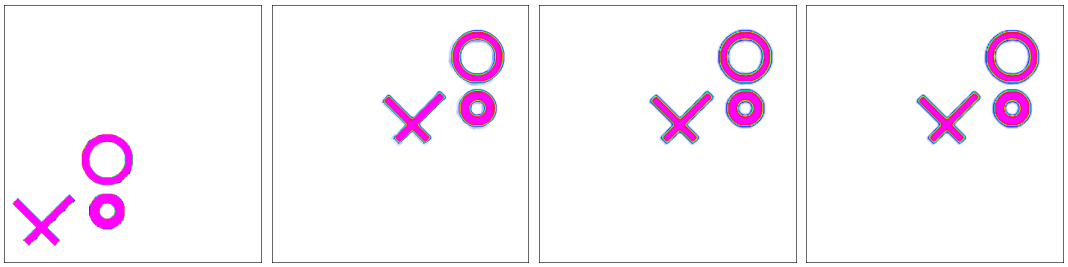
Cells	$\mathbb{Q}_1$ space	rate	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_3$ space	rate
128	7.43E-04		3.09E-04		4.63E-04	
256	1.76E-04	2.07	7.07E-05	2.12	1.59E-04	1.54
512	4.31E-05	2.03	1.60E-05	2.14	4.92E-05	1.69
1024	1.04E-05	2.04	3.47E-06	2.20	1.37E-05	1.84
2048	2.58E-06	2.01	7.28E-07	2.25	3.83E-06	1.83

(b) Edge-based FCT with the localized stencil N_i^*

Table 3.4: L^1 convergence of maximum principle preserving methods using continuous Galerkin finite elements for a smooth solution with local extrema.

3.5.4 Two dimensional advection with constant velocity field

Consider the discontinuous initial data shown in figure 3.6. In this problem the domain is given by $\Omega = (0, 100) \times (0, 100)$ and the velocity field by $v = (10, 10)$. The initial profile is, therefore, transported along the diagonal. We consider $\mathbb{Q}_1$ and $\mathbb{Q}_2$ spaces with number of cells adjusted to have 160801 degrees of freedom in all situations. We solve the problem using the edge-based FCT method with the full stencil N_i and the localized stencil N_i^*. The solution at $T = 4$ is shown in figure 3.6.



$(\min, \max) = (0, 1)$ $(7.97 \times 10^{-28}, 0.9984)$ $(3.23 \times 10^{-20}, 0.9978)$ $(1.4 \times 10^{-19}, 0.9928)$

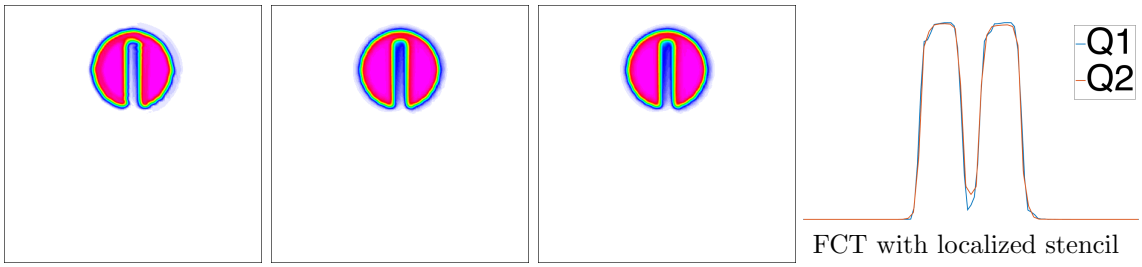
Figure 3.6: Two dimensional advection via different Flux Corrected Transport methods using continuous Galerkin finite elements. **Left:** Initial condition. **Middle-left:** standard/localized flux corrected transport method with $\mathbb{Q}_1$ space. **Middle-right:** standard flux corrected transport method with $\mathbb{Q}_2$ space. **Right:** localized flux corrected transport method with $\mathbb{Q}_2$ space. We adjust the number of cells between the spaces to have 160801 of freedom in all situations.

3.5.5 Two dimensional Zalesak disk

Now consider the test first proposed in [78]. The initial data is the characteristic function of a disc of radius $r = 0.15$ centered at the point $x_0 = (0.5, 0.75)$ with a thin rectangular area removed. The removed area is $\{x \in (x, y) \in \mathbb{R}^2 \mid |x - 0.5| < 0.025, y - 0.75 < 0.1125\}$. The velocity field is given by $\mathbf{v} = (-2\pi(y-0.5), 2\pi(x-0.5))$ which produces a rigid circular motion so that the exact solution coincides with the initial data at $T = 1$. We consider $\mathbb{Q}_1$ and $\mathbb{Q}_2$ spaces with number of cells adjusted to have 16641 degrees of freedom in all situations. We solve the problem using the edge-based FCT method with the full stencil N_i and the localize stencil N_i^*. The solution at $T = 1$ is shown in figure 3.7. In addition we show the solution (considering the localized stencil) along the cross section $y = 0.75$.

3.6 Conclusions

In this chapter we used the flux corrected transport method with high-order continuous Galerkin finite elements. We explored the behavior of two low-order methods



$(7.97 \times 10^{-28}, 0.9984)$ $(3.23 \times 10^{-20}, 0.9978)$ $(1.4 \times 10^{-19}, 0.9928)$

Figure 3.7: Two dimensional Zalesak disk via different Flux Corrected Transport methods using continuous Galerkin finite elements. **Left:** standard/localized flux corrected transport method with $\mathbb{Q}_1$ space. **Middle-left:** standard flux corrected transport method with $\mathbb{Q}_2$ space. **Middle-right:** localized flux corrected transport method with $\mathbb{Q}_2$ space. **Right:** plots at $y = 0.75$ comparing the localized flux corrected transport method with $\mathbb{Q}_1$ and $\mathbb{Q}_2$ spaces. We adjust the number of cells between the spaces to have 16641 of freedom in all situations.

under spaces of different order using positive and nodal basis functions. From these experiments we decided to use positive basis functions with the low-order method by [26]. This low-order method was used with a high-order continuous Galerkin discretization under the standard edge-based flux corrected transport methodology. We localize the bounds to solve the oscillatory behavior introduced with high-order spaces. See figure 3.4. This is the same idea as the method in §4.4 with discontinuous Galerkin finite elements. By using the localized bounds we highly reduced the oscillatory behavior but introduced high dissipation for higher-order spaces. See figure 3.5. We performed convergence tests for both methods (standard and localized bounds) and recover the expected high-order accuracy for smooth solutions that are monotone (using $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_3$). We observed no better than second-order (in the L^1 norm) with a smooth solution with local extrema for any space. This is a known problem for methods that impose monotonicity constraints.

4. FLUX CORRECTED TRANSPORT WITH DISCONTINUOUS GALERKIN FINITE ELEMENTS

We start this section motivated by the work in [2] where the authors apply the Flux Corrected Transport (FCT) method as revisited in §2.4. The spatial discretization is based on a discontinuous finite element space with positive shape functions given by Bernstein polynomials. The low-order method is an algebraic method given by [51,52] and the high-order method is based on Godunov (upwind) numerical flux. Under this setting the authors use up to $\mathbb{Q}_2$ spaces and obtain the expected third-order convergence and good quality results. However, if one applies the same methodology for higher order polynomials (around fourth and above) spurious oscillations are introduced. This is true even though the method is maximum principle preserving. The objective in this chapter is to explore alternatives to the FCT methodology to eliminate these non-physical oscillations under the same setting; i.e., using the same low- and high-order methods with the same finite element space and positive shape functions. This chapter is split as follows. We first revisit the high- and low-order methods in sections 4.1 and 4.2 respectively. Then we use the FCT method as in §2.4 and present results using polynomial spaces of different order where the non-physical oscillations are clear. Finally, in §4.4 and §4.5 we propose two methodologies to solve this problem.

4.1 High-order non-Maximum Principle Preserving method

4.1.1 Spatial discretization

We follow the spatial discretization of the high-order method in [2]. Consider a computational mesh $\mathcal{T}_h$ with internal faces $\mathcal{F}_h$. We define the discontinuous finite

dimensional space $X_h = \{\phi(\mathbf{x}) \in L^2(\Omega) : \phi|_K \in \mathbb{Q}|_K, \forall K \in \mathcal{T}_h\}$ where $\mathbb{Q}|_K$ is a polynomial space over the element K. Let $\{\phi_1, ..., \phi_N\}$ be a basis of X_h, where $N = \dim(X_h)$, such that $\sum_i \phi_i(\mathbf{x}) = 1$ and $\phi_i(\mathbf{x}) \geq 0, \forall \mathbf{x} \in \Omega$ for any $i = 1, ..., N$. We use Bernstein polynomials to construct such basis; moreover, we use this basis as shape functions for the finite element discretization.

Consider the transport equation (2.1) multiply it by $\phi \in X_h$, integrate over Ω and integrate by parts the advection term to obtain

$$\int_\Omega (\partial_t u)\phi d\mathbf{x} - \sum_{K\in\mathcal{T}_h} \int_K u(\mathbf{v}\cdot\nabla\phi)d\mathbf{x} + \sum_{f\in\mathcal{F}_h} \int_f u(\mathbf{v}\cdot\mathbf{n}_f)\phi d\mathbf{s} = 0, \tag{4.1}$$

where $\mathbf{s} \in \mathbb{R}^{d-1}$ and $\mathbf{n}_f$ is the unit normal vector at face f. Let $u_h \in X_h$ be the finite element approximation of u. Since u_h is discontinuous across f we can't replace u by u_h in (4.1) or we would obtain multiple values over f. Therefore, we define numerical fluxes associated with the internal faces to get

$$\int_\Omega (\partial_t u_h)\phi d\mathbf{x} - \sum_{K\in\mathcal{T}_h} \int_K u_h(\mathbf{v}\cdot\nabla\phi)d\mathbf{x} + \sum_{f\in\mathcal{F}_h} \int_f \{u_h\mathbf{v}\cdot\mathbf{n}_f\}_*[[\phi]]d\mathbf{s} = 0, \tag{4.2a}$$

where $[[\phi]] := \phi^- - \phi^+$, $\phi^\pm(\mathbf{x}) = \lim_{\xi\to 0^+} \phi(\mathbf{x} \pm \xi\mathbf{n}_f(\mathbf{x}))$ and

$$\{u_h\mathbf{v}\cdot\mathbf{n}_f\}_* = (\mathbf{v}\cdot\mathbf{n}_f)\left(\frac{u_h|_{K_1} + u_h|_{K_2}}{2}\right) + \frac{1}{2}|\mathbf{v}\cdot\mathbf{n}_f|[[u_h]], \tag{4.2b}$$

which is known as Godunov (upwind) flux [55, 67]. Method (4.2) can be recast in matrix-vector form as

$$M\frac{\partial U^H(t)}{\partial t} + TU^H(t) = 0, \tag{4.3a}$$

where $U^H(t) = [U_1^H(t), ..., U^H(t)_N]^t$ are the degrees of freedom of the finite element

solution $u_h(\mathbf{x}, t)$ at time t and M and T are the mass and transport matrices whose ij-th elements are given by

$$M_{ij} = \int_\Omega \phi_i \phi_j d\mathbf{x}, \tag{4.3b}$$

$$T_{ij} = -\int_\Omega \phi_j (\mathbf{v} \cdot \nabla \phi_i) d\mathbf{x} + \sum_{f \in \mathcal{F}_h} \int_f \{\phi_j \mathbf{v} \cdot \mathbf{n}_f\}_* [[\phi_i]] d\mathbf{s}. \tag{4.3c}$$

Remark 4.1.1.1 (Mass conservation)**.** *The method* (4.3) *is mass conservative in the following sense:*

$$\int_\Omega u_h(\mathbf{x}, t) d\mathbf{x} = \int_\Omega u_h(\mathbf{x}, t = 0) d\mathbf{x} \iff \sum_j U_j^H(t) m_j = \sum_j U_j^H(t = 0) m_j,$$

where $m_j = \sum_i \int_\Omega \phi_i \phi_j d\mathbf{x} = \int_\Omega \phi_j d\mathbf{x}$*. To see this consider the row sum of* (4.3)

$$\begin{aligned}
&\sum_i \sum_j M_{ij} \left(\frac{\partial U_j^H(t)}{\partial t} \right) = -\sum_i \sum_j T_{ij} U_j^H(t) \\
&\implies \sum_j m_j \left(\frac{\partial U_j^H(t)}{\partial t} \right) = -\sum_j U_j^H(t) \sum_i T_{ij} = 0 \\
&\implies \sum_j U_j^H(t) m_j = \sum_j U_j^H(t = 0) m_j,
\end{aligned}$$

where T *has zero column sum,*

$$\sum_i T_{ij} = -\int_\Omega \phi_j \left[\mathbf{v} \cdot \nabla \sum_i \phi_i \right] d\mathbf{x} + \sum_{f \in \mathcal{F}_h} \int_f \{\phi_j \mathbf{v} \cdot \mathbf{n}_f\}_* \left[\left[\sum_i \phi_i \right] \right] d\mathbf{s} = 0,$$

by partition of unity; i.e., $\sum_i \phi_i = 1$*.* □

4.1.2 Time discretization

For simplicity we present the full discretization considering Forward Euler integration in time. However, we extend the results to high-order approximations in time via Strong Stability Preserving (SSP) methods [20]. Moreover, all numerical experiments, unless otherwise noted, are performed via a third-order (with three stages) Runge-Kutta SSP method. The time discretization of (4.3) via Forward Euler is given by

$$M\left(\frac{U^H - U^n}{\Delta t}\right) + TU^n = 0, \tag{4.4}$$

where U^n and U^H are the degrees of freedom of the high-order finite element solution $u_h(\mathbf{x}, t)$ at time $t = t^n$ and $t = t^{n+1}$ respectively.

4.1.3 Numerical validation

In this section we perform a convergence test in space of the high-order method (4.4). To do this we consider a 1D problem over $\Omega = (0, 1) \subset \mathbb{R}$, with velocity $\mathbf{v} = 1$ and initial condition given by $u(\mathbf{x}, t = 0) = \cos(2\pi(x - 0.5))$. We use periodic boundary conditions and use the initial condition as exact solution at $T = 1$. In table 4.1 we show the results using different polynomial spaces. For these experiments we consider a fourth-order Runge-Kutta time integration.

4.2 Low-order Maximum Principle Preserving method

We consider the first-order MPP approach in [51, 52]. This method is based on applying a *discrete upwinding* to the transport matrix T of a high-order scheme and

Cells	$\mathbb{Q}_1$	rate	$\mathbb{Q}_2$	rate	$\mathbb{Q}_3$	rate
32	1.516E-03		1.943E-05		2.398E-07	
64	3.567E-04	2.087	2.421E-06	3.005	1.497E-08	4.002
128	8.633E-05	2.047	3.023E-07	3.001	9.355E-10	4.000
256	2.122E-05	2.024	3.778E-08	3.000	5.847E-11	4.000

Table 4.1: L^1 convergence of discontinuous Galerkin method (4.4) for a smooth solution with local extrema. We use a 4-th order Runge Kutta method.

lumping the mass matrix M. This leads to

$$M^L \left(\frac{U^L - U^n}{\Delta t} \right) + T^* U^n = 0, \tag{4.5a}$$

where U^n and U^L are the degrees of freedom of the low-order solution at time t^n and t^{n+1} respectively and M^L and T^* are the lumped mass matrix and the upwinded transport matrix respectively. They are given as follows:

$$M^L = M + L, \tag{4.5b}$$

$$T^* = T - D, \tag{4.5c}$$

where M and T are the consistent mass matrix and the transport matrix in the high-order method (4.4) and L and D are given by

$$L_{ij} = -M_{ij}, \qquad D_{ij} = \max(0, T_{ij}, T_{ji}), \tag{4.5d}$$

for the off-diagonal elements and

$$L_{ii} = -\sum_{j \neq i} L_{ij}, \qquad D_{ii} = -\sum_{j \neq i} D_{ij}, \tag{4.5e}$$

otherwise. Note that the elements of T^* are non-positive.

Remark 4.2.0.1 (The matrices L and $-D$ are diffusive matrices)**.** *The matrices L and D are symmetric, have non-positive off-diagonal entries and have zero row and column sum. These are the typical characteristics of the discretization of the Laplace operator $-\Delta$ [51].*

Remark 4.2.0.2 (Mass conservation)**.** *The method* (4.5) *is mass conservative in the following sense:*

$$\int_\Omega u_h(\mathbf{x},t)d\mathbf{x} = \int_\Omega u_h(\mathbf{x},t=0)d\mathbf{x} \iff \sum_i U_i^L m_i = \sum_i U_i^0 m_i,$$

where $m_i = \int_\Omega \phi_i(\mathbf{x})d\mathbf{x}$ is the i-th diagonal element of M^*. To see this consider the row sum of (4.5)*,*

$$\sum_i m_i \left(\frac{U_i^L - U_i^n}{\Delta t}\right) = -\sum_j U_j^n \sum_i (T_{ij} - D_{ij}) = 0 \implies \sum_i U_i^L m_i = \sum_i U_i^0 m_i,$$

where $\sum_i T_{ij} = 0$ from conservation of the high-order method (see remark 4.1.1.1).

□

Remark 4.2.0.3 (Maximum Principle Preserving)**.** *The method* (4.5) *is maximum principle preserving. To see this it is enough to show that for any $i = 1, ..., N$, U_i^L is a convex combination of U^n. We can rewrite method* (4.5) *as*

$$U_i^L = \sum_j R_{ij} U_j^n$$

where the off-diagonal entries of $R_{ij} = [(M^L)^{-1}(M^L - \Delta t T^*)]_{ij}$ are positive by the construction of M^L (given positive shape functions) and T^* and the diagonal ones

can be made positive via a CFL condition. Let $\mathbb{1}$ *be the vector of ones, then*

$$R\mathbb{1} = (M^L)^{-1}(M^L - \Delta t T^*)\mathbb{1} = (\mathbb{1} - \Delta t (M^L)^{-1} T^* \mathbb{1}) = \mathbb{1},$$

which is true since $T^*\mathbb{1} = (T + D)\mathbb{1} = 0$ *by conservation of the high-order method and the properties of the diffusive operator* D*. Therefore, for any* $i = 1, ..., N$, U_i^L *is a convex combination of* U^n*.* □

4.2.1 Numerical validation

We perform a convergence test in space for the low-order method (4.5). Consider $\Omega = (0, 1) \subset \mathbb{R}$ with velocity $\mathbf{v} = 1$ and initial condition given by $u(\mathbf{x}, t = 0) = \cos(2\pi(x - 0.5))$. We impose periodic boundary conditions and use the initial condition as exact solution at $T = 1$. In table 4.2 we show the results using different polynomial spaces. As expected, we obtain close to first-order convergence for all polynomial spaces. Observe that the error with the 23-rd order space is about 3 times smaller than with the 1-st order space; nevertheless, the number of degrees of freedom is vastly (12 times) larger. This indicates that the quality of the low-order method decreases as we consider higher-order spaces. We expect this to affect the quality of the FCT solution for high-order spaces. In the next section we reiterate on this problem.

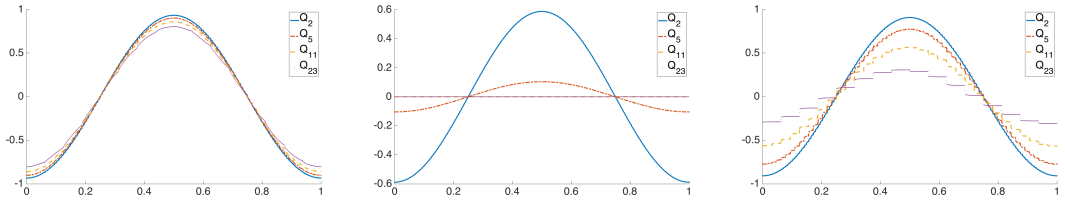
4.2.2 Low-order MPP method with positive v.s. non-positive basis functions

We explore the quality of the solution considering the 1D problem in §4.2.1 using positive and nodal basis functions. In particular, we consider Gauss-Legendre and Gauss-Lobatto nodal basis. The results are shown in figure 4.1. The solution is qualitatively similar for lower order polynomials. However, as we increase the order, the quality of the solution with nodal basis functions is highly reduced. With Gauss-

Cells	$\mathbb{Q}_1$	rate	$\mathbb{Q}_2$	rate	$\mathbb{Q}_3$	rate
32	1.708E-01		1.534E-01		1.385E-01	
64	9.163E-02	0.898	8.186E-02	0.906	7.340E-02	0.916
128	4.752E-02	0.947	4.231E-02	0.952	3.780E-02	0.957
256	2.420E-02	0.974	2.151E-02	0.976	1.918E-02	0.979
Cells	$\mathbb{Q}_5$	rate	$\mathbb{Q}_{11}$	rate	$\mathbb{Q}_{23}$	rate
32	1.189E-01		8.942E-02		6.585E-02	
64	6.247E-02	0.928	4.641E-02	0.946	3.383E-02	0.961
128	3.202E-02	0.964	2.364E-02	0.973	1.715E-02	0.981
256	1.622E-02	0.982	1.193E-02	0.987	8.630E-03	0.990

Table 4.2: L^1 convergence of edge based low-order method (4.5).

Legendre the solution is extremely dissipated for the larger order polynomials. With Gauss-Lobatto the solution is also more dissipated than if positive basis functions are used but not as much as with Gauss-Legendre; however, the solution is less smooth than before.

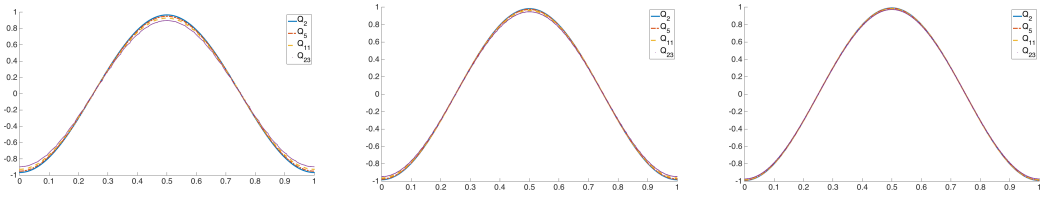


(a) Positive basis functions (b) Gauss-Legendre nodal basis (c) Gauss-Lobatto nodal basis

Figure 4.1: Low-order method with positive v.s. nodal basis functions using discontinuous Galerkin fintie elements. We consider the low-order method (4.5) and use $\mathbb{Q}_2$, $\mathbb{Q}_5$, $\mathbb{Q}_{11}$ and $\mathbb{Q}_{23}$ spaces. The number of cells is adjusted to have 384 degrees of freedom in all simulations.

It is also important to note that even with positive basis functions the solution is more dissipated as we increase the order of the polynomials. To further study this, we consider the same experiment with the same polynomial spaces but increase the

number of degrees of freedom. In figure 4.2 we show the solution with 768, 1536 and 3072 degrees of freedom. In all situations we obtained more dissipated solutions as the order of the polynomial space is increased. However, the solutions get closer as we refine the mesh. In table 4.2 we can see that the convergence rate is slightly higher as the order is increased. From these two results we expect that using higher-order polynomials eventually gives better results. However, the resolution needed might be too large. As mentioned in the previous section, this is important to consider when the low-order method is used within the FCT methodology.



(a) 768 degrees of freedom (b) 1536 degrees of freedom (c) 3072 degrees of freedom

Figure 4.2: Low-order method with positive basis functions using discontinuous Galerkin fintie elements with multiple refinements. We consider the low-order method (4.5) and use $\mathbb{Q}_2$, $\mathbb{Q}_5$, $\mathbb{Q}_{11}$ and $\mathbb{Q}_{23}$ spaces. The number of cells is adjusted to have (a) 768, (b) 1536 and (c) 3072 degrees of freedom.

4.3 Edged-based Flux Corrected Transport with discontinuous Galerkin finite elements

In this section we use the standard edge-based FCT methodology revisited in §2.4 using the low- and high-order methods (4.5) and (4.4) respectively. We consider first

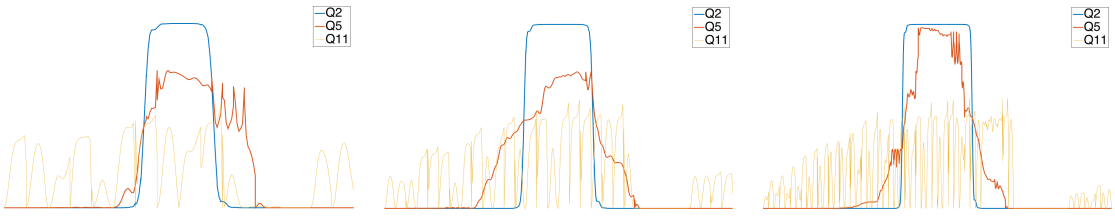
1D simulations with discontinuous initial data given by

$$u_h(\mathbf{x}, t=0) = \begin{cases} 1, & \forall x \in (0.4, 0.6) \\ 0, & \text{otherwise} \end{cases} \tag{4.6}$$

over $\Omega = (0,1) \subset \mathbb{R}$ and velocity given by $\mathbf{v} = 1$. The initial condition is obtained as explained in §2.2. We use $\mathbb{Q}_2$, $\mathbb{Q}_5$ and $\mathbb{Q}_{11}$ spaces and consider different refinements. For each refinement the number of cells is adjusted to have the same number of degrees of freedom between all spaces. We impose periodic boundary conditions and show the solution at $t = 1$. The results are shown in figure 4.3. In addition, we consider a more complicated discontinuous initial condition in 2D, shown in figure 4.4, with $\Omega = [0, 100] \times [0, 100] \subset \mathbb{R}^2$ and velocity $\mathbf{v} = (10, 10)$. We compute the solution at time $t = 4$ using $\mathbb{Q}_2$ and $\mathbb{Q}_5$ spaces with the number of cells adjusted to have the same number of degrees of freedom in both spaces. The results are shown in figure 4.4. The problem is clear, as we consider higher-order spaces large non-physical oscillations are introduced making the solution unacceptable. This is true nevertheless the solution is maximum principle preserving. In the remainder of this chapter we propose two approaches to overcome this issue.

4.4 Localized Flux Corrected Transport with discontinuous Galerkin finite elements

The FCT method in §2.4 is local in the sense given by the sparsity pattern of the transport matrix T. For discontinuous Galerkin finite elements this sparsity pattern includes all degrees of freedom on a given cell and on cells sharing a face with it, see figure 4.5a. When the order of the polynomial space is small, the sparsity pattern includes few degrees of freedom; however, as we increase the order, the number of degrees of freedom in the sparsity pattern increases. The method *loses* locality with



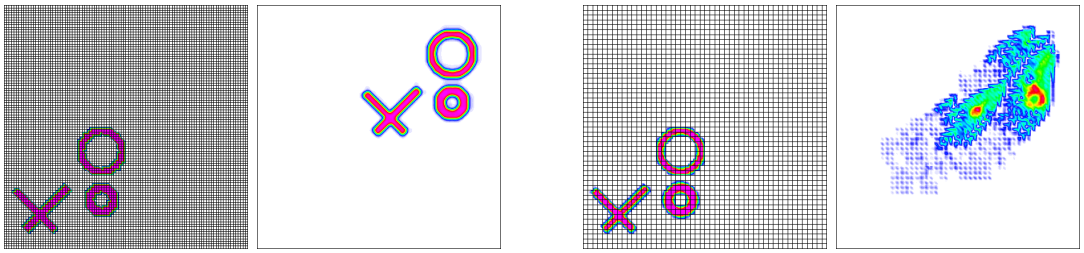
192 degrees of freedom 384 degrees of freedom 768 degrees of freedom

Figure 4.3: Standard Flux Corrected Transport method on a 1D problem with discontinuous initial condition using discontinuous Galerkin finite elements. The method is given by (2.8) with the low- and high-order methods in sections §4.2 and §4.1 respectively. We consider different polynomial spaces and multiple refinements. For each refinement the number of cells is adjusted to have the same number of degrees of freedom in all spaces.

respect to degrees of freedom; nevertheless, it is fixed with respect to number of cells. In an extreme case we can consider a single cell with a polynomial space of order as large as needed to have roughly certain number of degrees of freedom. In this situation, being *on bounds* means nothing since the sparsity pattern includes all degrees of freedom in the finite element space. This motivates the idea of considering tighter bounds. We propose to localize the bounds by mimicking the stencil of a first-order space; i.e., for a given i-th degree of freedom we consider those at locations adjacent to i. Let N_i be the conventional neighborhood for the i-th degree of freedom. For the finite element space $\mathbb{Q}_k$, we use tighter bounds given by the stencil

$$N_i^* = \left\{ j \in N_i : \ \operatorname{dist}(i,j) \leq \frac{\sqrt{d}}{k} \right\}, \tag{4.7}$$

where $d \in \{1,2,3\}$ is the space dimension and $\operatorname{dist}(i,j)$ is the Euclidean distance between the i-th and j-th degrees of freedom's images on the reference element. Defining N_i^* with respect to the reference element makes the approach applicable to unstructured grids. In figure 4.5 we consider a representative degree of freedom



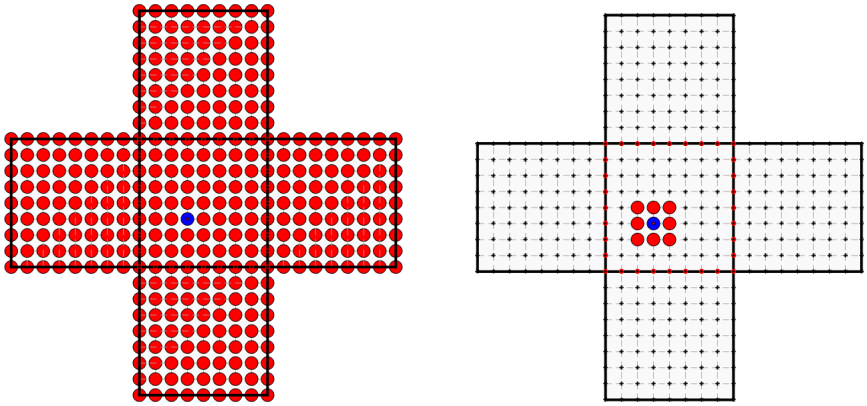
$\mathbb{Q}_2$, $(\min, \max) = (-2.587 \times 10^{-15}, 1.99)$ $\mathbb{Q}_5$, $(\min, \max) = (-1.363 \times 10^{-14}, 1.632)$

Figure 4.4: Standard Flux Corrected Transport method on a 2D problem with discontinuous initial condition using discontinuous Galerkin finite elements. The method is given by (2.8) with the low- and high-order methods given by (4.5) and (4.4) respectively. We use $\mathbb{Q}_2$ and $\mathbb{Q}_5$ spaces with the number of cells adjusted so that 90000 degrees of freedom are used in both simulations. For each case, we show (left) the initial condition with the grid and (right) the solution at $t = 4$.

in thick blue and show the conventional or full stencil via the sparsity pattern of the transport matrix T. In addition, we show the tighter bounds (4.7), mimicking a stencil of a first-order space. Note that since DG discretization is used we have two degrees of freedom at the faces. This is denoted by using a red circle and a black cross in those locations.

Remark 4.4.0.1 (Low-order method is non Maximum Principle Preserving in the tighter bounds)**.** *The low-order method* (4.5) *is guaranteed to produce a MPP solution on the conventional bounds; i.e., including all degrees of freedom in the sparsity pattern of* T*. Since the tighter bounds consider a smaller set of degrees of freedom there is no guarantee the low-order solution is MPP in this set.*

Due to remark 4.4.0.1, we can't use the standard FCT methodology with the bounds in (2.2) given by the tighter stencil (4.7). To overcome this, we modify the



(a) Full stencil (b) First-order stencil

Figure 4.5: Full and localized stencil to compute bounds using discontinuous spaces. In (a) we show the conventional or full stencil in a DG discretization. In (b) we mimic a $\mathbb{Q}_1$ space. The thick blue mark represents the degree of freedom for which we compute the bounds. The thick red marks represent the degrees of freedom included to compute the bounds. The non-thick black dots in (b) represent all degrees of freedom in the sparsity pattern of T and the non-think red marks in (b) indicate a double degree of freedom (due to DG finite element spaces), which are also considered for the computation of the bounds.

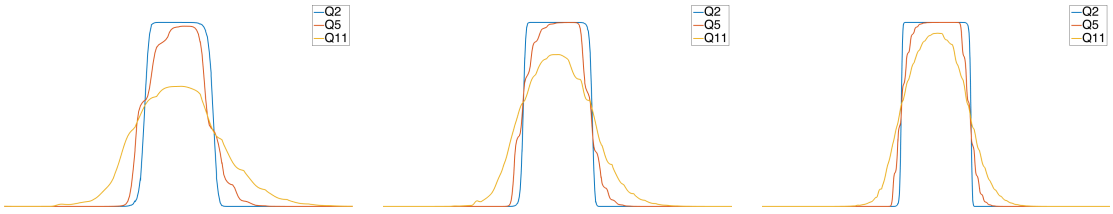
bounds in (2.2) to be

$$U_i^{\min} = \min\left(U_i^L, \min_{j\in N_i^*} U_j^n\right), \tag{4.8a}$$

$$U_i^{\max} = \max\left(U_i^L, \max_{j\in N_i^*} U_j^n\right), \tag{4.8b}$$

which guarantees the low-order solution is on bounds and, therefore, we can apply the FCT methodology. We now repeat the simulations in §4.3 using the tighter bounds (4.8). The results are shown in figures 4.6 and 4.7. We observe the oscillatory behavior, although not completely eliminated, is highly reduced. It is clear also the high amount of dissipation introduced as the order of the polynomial is increased. In the next section we propose a FCT-*like* methodology that reduces the oscillatory

behavior even more and yields less dissipated solutions.



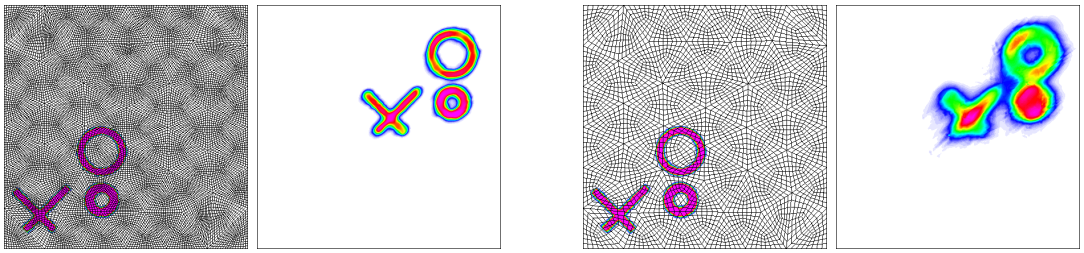
192 DOFs 384 DOFs 768 DOFs

Figure 4.6: Localized Flux Corrected Transport method on a 1D problem with discontinuous initial condition using discontinuous Galerkin finite elements. We use the low- and high-order methods (4.5) and (4.4) respectively. We consider different polynomial spaces and multiple refinements. For each refinement the number of cells is adjusted to have the same number of degrees of freedom in all spaces.

4.5 Element based Flux Corrected Transport with discontinuous Galerkin finite elements

In the standard FCT method revisited in §2.4 we start with two methods that are mass conservative. One is low-order and MPP and the other is high-order but non-MPP. Then, an interpolation is made from the low- to the high-order solution to obtain a solution that is MPP. For any degree of freedom U_i^{n+1} there are as many interpolating parameters as degrees of freedom in the support of U_i^{n+1}. These interpolating parameters are designed to preserve conservation of mass.

In this section we introduce a FCT-*like* method that considers two MPP solutions. One is low-order and mass conservative and the other is (presumably) high-order but non mass conservative. Then, we interpolate from the low- to the high-order solution to recover mass conservation cell-wise. For any degree of freedom U_i^{n+1} we have just one interpolating parameter. This interpolating parameter is designed to maintain



$\mathbb{Q}_2$, $(\min, \max) = (-2.309 \times 10^{-17}, 1.976)$ $\mathbb{Q}_5$, $(\min, \max) = (-1.686 \times 10^{-20}, 1.322)$

Figure 4.7: Localized Flux Corrected Transport method on a 2D problem with discontinuous initial condition using discontinuous Galerkin finite elements. The bounds are given by (4.8). The low- and high-order methods are given by (4.5) and (4.4) respectively. We use $\mathbb{Q}_2$ and $\mathbb{Q}_5$ spaces with the number of cells adjusted so that 127872 degrees of freedom are used in both simulations. For each case we show (left) the initial condition with the grid and (right) the solution at $t = 4$.

the solution on bounds. It is important to emphasize that the recovery on mass conservation is obtained per cell and not globally. Moreover, we propose a methodology to localize even more this redistribution of mass inside a cell. Recovering the conservation of mass within a cell is possible due to local mass properties of the low- and high-order methods we consider in this work and by using discontinuous Galerkin finite elements. We explain this in more detail in the next section.

4.5.1 Mass conservation of low- and high-order methods

In this section we show that the low- and the high-order solutions in (4.5) and (4.4) have the same mass on any given cell $K \in \mathcal{T}_h$; i.e., $\sum_{i \in \mathcal{I}_K} m_i U_i^H = \sum_{i \in \mathcal{I}_K} m_i U_i^L$, where $\mathcal{I}_K$ is the index set of all degrees of freedom in cell K. This is true by using the low- and high-order methods with discontinuous Galerkin finite elements described in §4.2 and §4.1 and the positive shape functions described in §4.1.1. First,

we rewrite the high-order method as

$$m_i(U_i^H - U_i^L) = f_i^H, \tag{4.9a}$$

where f_i^H is a high-order flux correction given by

$$f_i^H = \sum_j (M^L - M)_{ij}\delta U_j + \Delta t \sum_j D_{ij} U_j^n. \tag{4.9b}$$

Here $\delta U_j := U_j^H - U_j^n$. Then we use the following:

Lemma 4.5.1.1 (Flux correction is massless on any cell)**.** *Using positive shape functions and the low- and high-order methods with discontinuous Galerkin finite elements described in §4.2 and §4.1, the flux correction* f^H (4.9b) *is massless on any cell; i.e.,* $\sum_{i\in\mathcal{I}_K} f_i^H = 0, \forall K \in \mathcal{T}_h$.

Proof. Given any cell $K \in \mathcal{T}_h$ consider

$$\sum_{i\in\mathcal{I}_K} f_i^H = \sum_j \delta U_j \sum_{i\in\mathcal{I}_K} (M^L - M)_{ij} + \Delta t \sum_j U_j \sum_{i\in\mathcal{I}_K} D_{ij}.$$

It is our aim to show that $\sum_{i\in\mathcal{I}_K} f_i^H = 0$. Since we use a DG discretization, any shape function is supported on a single cell. Therefore, $\forall i \in [1,\dots,N] \quad \sum_{j=1,\dots,N} M_{ij}^L = \sum_{j\in\mathcal{I}_K} M_{ij}^L = M_{ii}^L = \sum_{j=1,\dots,N} M_{ij} = \sum_{j\in\mathcal{I}_K} M_{ij}$ and, hence, $\sum_{i\in\mathcal{I}_K}(M^L - M)_{ij} = 0$.

From remark 4.2.0.1, we recall that $\sum_{i\in[1,\dots,N]} D_{ij} = 0$. Now we show that $D_{ij} = 0$ whenever i and j belong to different cells. Recall the definition of D_{ij} from (4.5):

$$D_{ij} = \max(0, T_{ij}, T_{ji}), \quad \forall i \neq j \tag{4.10}$$

and $D_{ij} = -\sum_{j\neq i} D_{ij}$, otherwise. We just need to consider the off-diagonal elements

and assume they belong to different cells. From §4.1, the i, j-th element of the transport matrix is given by

$$T_{ij} = -\int_{\Omega} \phi_j(\mathbf{v}\cdot\nabla\phi_i)d\mathbf{x} + \sum_{f\in\mathcal{F}_h}\int_f \{\phi_j\mathbf{v}\cdot\mathbf{n}_f\}_*[[\phi_i]]d\mathbf{s}, \tag{4.11}$$

where the first integral is zero since i and j belong to different cells and each shape function is supported on its corresponding cell. Recall the definition of the numerical flux:

$$\{\phi\mathbf{v}\cdot\mathbf{n}_F\}_* = (\mathbf{v}\cdot\mathbf{n}_F)\left(\frac{\phi|_{K_1}+\phi|_{K_2}}{2}\right) + \frac{1}{2}|\mathbf{v}\cdot\mathbf{n}_F|[[\phi]], \tag{4.12a}$$

$$[[\phi]] = \phi^- - \phi^+, \tag{4.12b}$$

$$\phi^{\pm}(\mathbf{x}) = \lim_{\xi\to 0^+}\phi(\mathbf{x}\pm\xi\mathbf{n}_f(\mathbf{x})). \tag{4.12c}$$

We choose a definition for the normal vector $\mathbf{n}_F$ to go from cell K_1 to cell K_2. Using this definition we get $[[\phi]] = \phi|_{K_1} - \phi|_{K_2}$. Assume ϕ_j is supported on cell K_1 and ϕ_i on cell K_2, then $[[\phi_j]] = \phi_j|_{K_1}$ and $[[\phi_i]] = -\phi_i|_{K_2}$, which leads to

$$\{\phi_j\mathbf{v}\cdot\mathbf{n}_F\}_*[[\phi_i]] = -\left[|\mathbf{v}\cdot\mathbf{n}_F| + (\mathbf{v}\cdot\mathbf{n}_F)\right]\left(\frac{\phi_j|_{K_1}\phi_i|_{K_2}}{2}\right),$$

which is non-positive regardless of the sign of $\mathbf{v}\cdot\mathbf{n}_F$ provided the shape functions are positive. Similarly, if ϕ_j is supported on cell K_2 and ϕ_i on cell K_1, then $[[\phi_j]] = -\phi_j|_{K_2}$ and $[[\phi_i]] = \phi_i|_{K_1}$, which leads to

$$\{\phi_j\mathbf{v}\cdot\mathbf{n}_F\}_*[[\phi_i]] = \left[(\mathbf{v}\cdot\mathbf{n}_F) - |\mathbf{v}\cdot\mathbf{n}_F|\right]\left(\frac{\phi_j|_{K_2}\phi_i|_{K_1}}{2}\right),$$

which is also non-positive provided we use positive shape functions. Therefore, $T_{ij} \leq$

0 whenever i and j don't belong to the same cell. This implies that $D_{ij} = 0$ whenever i and j don't belong to the same cell.

From $\sum_{i\in[1,\dots,N]} D_{ij} = 0$ and $D_{ij} = 0$ whenever i and j don't belong to the same cell we conclude that $\sum_{i\in\mathcal{I}_K} D_{ij} = 0$. Therefore, we get

$$\sum_{i\in\mathcal{I}_K} f_i^H = \sum_j \delta U_j \sum_{i\in\mathcal{I}_K} (M^L - M)_{ij} + \Delta t \sum_j U_j \sum_{i\in\mathcal{I}_K} D_{ij} = 0.$$

□

Then, from (4.9a), it is clear that $\sum_{i\in\mathcal{I}_K} m_i U_i^H = \sum_{i\in\mathcal{I}_K} m_i U_i^L$; i.e., the low- and the high-order methods have the same mass on any cell $K \in \mathcal{T}_h$. This allow us to consider non mass conservative flux corrections that assure the solution is on bounds and then adjust those fluxes to recover mass conservation per cell. To do this we need to adjust the fluxes on any cell without modifying fluxes in other cells. This is possible since we consider discontinuous Galerkin finite elements.

4.5.2 Clipped solution

The first stage of this method is to clip the solution considering some local bounds. We can consider different options depending on the stencil; i.e., we can consider the full or conventional stencil N_i (figure 4.5a) or the tighter stencil N_i^* from equation (4.7) (figure 4.5b). In either case we compute the bounds via (2.2) to obtain $U_i^{\min}$ and $U_i^{\max}$. Then we consider the high-order solution U^H from method (4.4) to get

$$U_i^* = \min(U_i^{\max}, \max(U_i^H, U_i^{\min})) \tag{4.13}$$

where U_i^* is the clipped solution.

In figure 4.8 we consider the 1D problem with discontinuous initial data and show the results of clipping the solution with the bounds computed via the full and the

tighter stencil. We use $\mathbb{Q}_5$ and $\mathbb{Q}_{11}$ spaces. It is clear that non-physical oscillations are present when the full stencil is considered. For this reason we compute the bounds (2.2) using the tighter stencil N_i^* (4.7). Finally, in figure 4.9 we show the results of the same problem considering different spaces and different refinements. Two observations can be made. First, phase errors appear due to not conserving mass. Mass conservation is addressed in the next section. Second, the clipped solution becomes more dissipated as one considers higher order spaces.

Remark 4.5.2.1 (Clipped solution is MPP in the tighter and the full bounds)**.** *The clipped solution U_i^* using the tighter stencil is MPP in such stencil; i.e.,*

$$\min_{j\in N_i^*} U_j^n =: U_i^{min} \leq U_i^* \leq U_i^{max} := \max_{j\in N_i^*} U_j^n,$$

where N_i^* denotes the tighter stencil. Being on bounds in $N_i^* \subset N_i$ implies being on bounds in any larger set of degrees of freedom; in particular, in the conventional or full stencil N_i; i.e.,

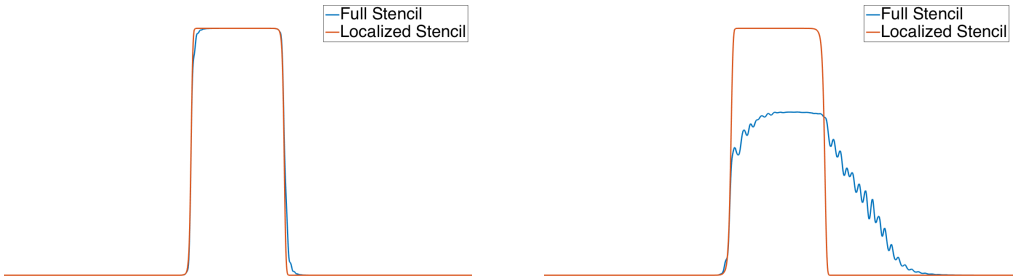
$$\min_{j\in N_i} U_j^n =: U_i^{min} \leq U_i^* \leq U_i^{max} := \max_{j\in N_i} U_j^n.$$

Therefore, the clipped solution is on bounds in both the tighter and the full stencil.

4.5.3 Local recovery on mass conservation

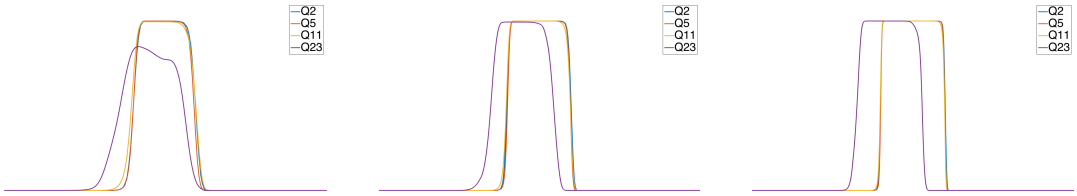
In this section we consider the clipped solution U_i^* and recover mass conservation per cell. In §4.5.1 we saw that the high-order flux correction

$$f_i^H = m_i(U_i^H - U_i^L) \tag{4.14}$$



(a) $\mathbb{Q}_5$ (b) $\mathbb{Q}_{11}$

Figure 4.8: Clipped solution via the full stencil v.s. the localized stencil on a 1D problem with discontinuous initial condition using discontinuous Galerkin finite elements. The method is given by (4.13). We use $\mathbb{Q}_5$ and $\mathbb{Q}_{11}$ spaces. The number of cells is adjusted so that 768 degrees of freedom are used in both simulations.



(a) 192 DOFs (b) 384 DOFs (c) 768 DOFs

Figure 4.9: Clipped solution via the localized stencil on a 1D problem with discontinuous initial condition using discontinuous Galerkin finite elements. The method is given by (4.13). We use $\mathbb{Q}_2$, $\mathbb{Q}_5$, $\mathbb{Q}_{11}$ and $\mathbb{Q}_{23}$ spaces. The number of cells is adjusted to have (a) 192, (b) 384 and (c) 768 degrees of freedom.

has zero mass within a cell; i.e., $\sum_{i\in\mathcal{I}_K} f_i^H = 0, \forall K \in \mathcal{T}_h$. Given the clipped solution U_i^*, define

$$f_i^* := m_i(U_i^* - U_i^L). \tag{4.15}$$

Here f_i^* is a flux correction from the low-order to the clipped solution. To recover mass conservation per cell we need to modify the fluxes $f_i^* \mapsto f_i$ so that $\sum_{i\in\mathcal{I}_K} f_i =$

$0, \forall K \in \mathcal{T}_h$. The modification of the fluxes has to be done without creating violations on the maximum principle; i.e., the solution must remain on bounds.

4.5.3.1 Mass conservation via flux scaling

Let $0 \leq \alpha_i \leq 1$ and define U_i^{n+1} via

$$m_i(U_i^{n+1} - U_i^L) = \alpha_i f_i^* \tag{4.16}$$

where α_i's are defined so that $\sum_{i \in \mathcal{I}_K} \alpha_i f_i^* = 0, \forall K \in \mathcal{T}_h$. Note that this is always possible. In particular, one might choose $\alpha_i = 0$ which gives back the low-order solution.

Remark 4.5.3.1 (Mass conservation)**.** *Assuming the low-order solution is mass conservative; i.e.,* $\sum_i m_i U_i^L = \sum_i m_i U_i^0$ *and given* $\sum_{i \in \mathcal{I}_K} \alpha_i f_i^* = 0, \forall K \in \mathcal{T}_h$*, then the method* (4.16) *is mass conservative.*

To see this consider $\sum_{i \in \mathcal{I}_K} \alpha_i f_i^* = 0, \forall K \in \mathcal{T}_h \implies \sum_i \alpha_i f_i^* = 0$*. Therefore,* $\sum_i m_i U_i^{n+1} = \sum_i m_i U_i^L$*. By mass conservation of the low-order method we get* $\sum_i m_i U_i^{n+1} = \sum_i m_i U_i^0 \implies \int_\Omega u_h(\mathbf{x}, t) d\mathbf{x} = \int_\Omega u_h(\mathbf{x}, 0) d\mathbf{x}$*.* □

Theorem 4.5.3.1 (Maximum-Principle Preservation (MPP))**.** *Given* $0 \leq \alpha_i \leq 1$ *and provided* U_i^* *and* U_i^L *are on bounds; i.e.,* $U_i^{\min} \leq U_i^{*/L} \leq U_i^{\max}$*, the method* (4.16) *is MPP; i.e.,* $U_i^{\min} \leq U_i^{n+1} \leq U_i^{\max}$*, where* $U_i^{\min} := \min_{j \in N_i} U_j^n$ *and* $U_i^{\max} := \max_{j \in N_i} U_j^n$*.*

Proof. Rewrite method (4.16) as

$$U_i^{n+1} = U_i^L + \alpha_i m_i^{-1} f_i^* = U_i^L + \alpha_i (U_i^* - U_i^L) = \alpha U_i^* + (1 - \alpha_i) U_i^L.$$

Since $0 \leq \alpha_i \leq 1$ and $U_i^{*/L} \leq U_i^{\max}$, we get

$$U_i^{n+1} \leq \alpha_i U_i^{\max} + (1-\alpha_i) U_i^{\max} = U_i^{\max} \implies U_i^{n+1} \leq U_i^{\max}.$$

The lower bound is proven similarly. □

There are different strategies to choose the interpolating parameters. A first approach, which we refer as uniform mass-distribution, is to scale down the dominant fluxes by the same factor. Consider a representative cell $K \in \mathcal{T}_h$, let $\mathcal{I}_K$ denote the index set of all degrees of freedom on cell K and define

$$S_K^+ = \sum_{\substack{f_i^* > 0 \\ i \in \mathcal{I}_K}} f_i^*, \quad S_K^- = \sum_{\substack{f_i^* < 0 \\ i \in \mathcal{I}_K}} f_i^*.$$

If $S_K^+ + S_K^- > 0,\ i \in \mathcal{I}_K$ we choose

$$\alpha_i := \begin{cases} -\frac{S^-}{S^+} & \text{if } f_i^* > 0 \\ 1 & \text{otherwise} \end{cases}. \tag{4.17a}$$

If $S_K^+ + S_K^- < 0,\ i \in \mathcal{I}_K$ we choose

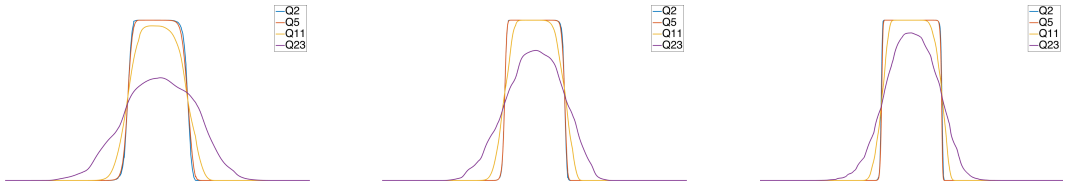
$$\alpha_i := \begin{cases} 1 & \text{if } f_i^* \geq 0 \\ -\frac{S^+}{S^-} & \text{otherwise} \end{cases}, \tag{4.17b}$$

and if $S_K^+ + S_K^- = 0,\ i \in \mathcal{I}_K$ then $\alpha_i = 1$. It is easy to see that $0 \leq \alpha_i \leq 1$ and $\sum_{i \in \mathcal{I}_K} \alpha_i f_i^* = 0, \forall K \in \mathcal{T}_h$.

In figure 4.10 we show results of the 1D problem with discontinuous initial data considering the uniform mass-distribution method (4.16), (4.17) with $\mathbb{Q}_2$, $\mathbb{Q}_5$, $\mathbb{Q}_{11}$

and $\mathbb{Q}_{23}$ spaces. We consider multiple refinements and for each adjust the number of cells to have the same number of degrees of freedom between all spaces. We can easily identify a problem, namely, the solution is more dissipated as we consider higher-order spaces. We recall from figures 4.2 and 4.9 that the low-order method and the clipping process produce more dissipative results as we increase the order. This is part of the problem. In addition, the redistribution of mass to preserve mass conservation is introducing more dissipation as the order is increased.

The recovery of mass conservation is performed cell-wise. When higher-order spaces are used, more degrees of freedom have to be considered within a cell to recover mass conservation. Therefore, locality is lost with respect to degrees of freedom. Motivated by this, in the following section, we propose a process to recover mass conservation that is more localized; i.e., the distribution of mass is performed differently in different parts of a given cell.



(a) 192 DOFs (b) 384 DOFs (c) 768 DOFs

Figure 4.10: Element based Flux Corrected Transport with uniform mass-distribution on a 1D problem with discontinuous initial condition. The method is given by (4.16), (4.17). We consider different polynomial spaces and multiple refinements. For each refinement the number of cells is adjusted to have the same number of degrees of freedom in all spaces.

4.5.3.2 Mass conservation via penalization

In this section the recovery of mass conservation is done at a sub-cell level. We discuss two possible strategies.

In [16] a *repair* is done to obtain a MPP mass conservative method. This repair considers a given cell and those adjacent to it. If a criteria-satisfying solution cannot be found, more cells are considered until a mass conservative solution that is on bounds is obtained. We can apply this idea to redistribute the mass restricted to a cell; i.e., consider a degree of freedom within a cell and try to distribute the mass considering just adjacent degrees of freedom in such a way that the mass for this set of degrees of freedom equals the mass of the high-order flux f^H on the set. If that is not possible without violating the maximum principle, we consider a larger set. In the worst case scenario, we would have to consider the entire cell and use an approach similar to that presented in §4.5.3.1.

Another approach to distribute the mass within a cell is as follows. In [46], the author obtains a solution on bounds by clipping the solution and doing a global fix in mass using a Lagrange multiplier. We use that same idea but restricted to a single cell. Doing so we get

$$m_i(U_i^{n+1} - U_i^L) = f_i^* - \lambda_K z_i, \tag{4.18a}$$

where z_i is a penalization term and λ_K is given by

$$\lambda_K = \frac{\sum_{i\in\mathcal{I}_K} f_i^*}{\sum_{i\in\mathcal{I}_K} z_i}. \tag{4.18b}$$

Remark 4.5.3.2 (Mass conservation)**.** *Given any choice of z_i's provided that at least one is different than zero at every cell and that the low-order method is mass*

conservative; i.e., $\int_\Omega u_h^L(\mathbf{x},t)d\mathbf{x} = \int_\Omega u_h(\mathbf{x},0)d\mathbf{x} \implies \sum_i m_i U_i^L = \sum_i m_i U_i^0$ *then the method* (4.18) *is mass conservative; i.e.,* $\int_\Omega u_h(\mathbf{x},t)d\mathbf{x} = \int_\Omega u_h(\mathbf{x},0)d\mathbf{x}$.

If at least one $z_i \neq 0$ *then* λ_K *is properly defined. Consider* $\sum_{i\in\mathcal{I}_K} m_i(U_i^{n+1} - U_i^L) = \sum_{i\in\mathcal{I}_K} f_i^* - \lambda_K \sum_{i\in\mathcal{I}_K} z_i = 0 \implies \sum_i m_i U_i^{n+1} = \sum_i m_i U_i^L$. *By conservation of mass of the low-order solution we get* $\sum_i m_i U_i^{n+1} = \sum_i m_i U_i^0 \implies \int_\Omega u_h(\mathbf{x},t)d\mathbf{x} = \int_\Omega u_h(\mathbf{x},0)d\mathbf{x}$. □

Theorem 4.5.3.2 (Maximum-Principle Preserving (MPP))**.** *Let* $\delta_K := \sum_{i\in\mathcal{I}_K} f_i^*$. *Assume the low-order solution* U_i^L *and the clipped solution* U_i^* *are on bounds; i.e.,* $U_i^{\min} \leq U_i^{*/L} \leq U_i^{\max}$ *and that the penalization terms* z_i*'s satisfy the following conditions:*

(1). If $\delta_K = 0$ *then* $z_i = 0$,

(2). If $\delta_K > 0$ *and* $f_i^* \leq 0$ *then* $z_i = 0$,

(3). If $\delta_K < 0$ *and* $f_i^* \geq 0$ *then* $z_i = 0$,

(4). $Sign(z_i) = Sign(f_i^*)$,

(5). $\lambda_K z_i / f_i^* \leq 1$,

then the method (4.18) *is MPP; i.e.,* $U_i^{\min} \leq U_i^{n+1} \leq U_i^{\max}$, *where* $U_i^{\min} := \min_{j\in N_i} U_j^n$ *and* $U_i^{\max} := \max_{j\in N_i} U_j^n$.

Proof. Assume $\delta_K = 0$. By assumption (1) $z_i = 0$, then $m_i(U_i^{n+1} - U_i^L) = f_i^* = m_i(U_i^* - U_i^L) \implies U_i^{n+1} = U_i^*$ and since U_i^* is on bounds $U_i^{\min} \leq U_i^{n+1} \leq U_i^{\max}$.

Assume $\delta_K > 0$. If $f_i^* \leq 0$, by assumption (2), $z_i = 0 \implies U_i^{n+1} = U_i^*$ and since U_i^* is on bounds $U_i^{\min} \leq U_i^{n+1} \leq U_i^{\max}$. If $f_i^* > 0$, by assumption (4) and using the

definition of λ_K we get $\lambda_K z_i > 0$ and then

$$m_i(U_i^{n+1} - U_i^L) = f_i^* - \lambda_K z_i \leq f_i^* = m_i(U_i^* - U_i^L) \implies U_i^{n+1} \leq U_i^* \leq U_i^{\max}$$
$$\implies U_i^{n+1} \leq U_i^{\max}.$$

For the lower bound consider assumption (5) to get

$$m_i(U_i^{n+1} - U_i^L) = f_i^* - \lambda_K z_i \geq 0 \implies U_i^{n+1} \geq U_i^L \geq U_i^{\min} \implies U_i^{n+1} \geq U_i^{\min}.$$

If $\delta_K < 0$ we proceed similarly but using assumption (3) instead of (2). □

Remark 4.5.3.3 (Mass-conservation when $\delta_K = \sum_{i \in \mathcal{I}_K} f_i^* = 0$)**.** *To obtain a solution on bounds if $\delta_K = 0$ we choose $z_i = 0$ in cell K. In this situation λ_K in* (4.18b) *is not properly defined and the assumptions of remark 4.5.3.2 are not satisfied. However, if $\delta_K = 0$ the corresponding flux correction is massless and it doesn't contribute on loosing mass conservation.*

Remark 4.5.3.4 (Mass-recovery via flux scaling v.s. flux penalization)**.** *Both approaches to recover mass conservation are equivalent. This can be seen by choosing the scaling factor in* (4.16) *to be*

$$\alpha_i = \begin{cases} 1 - \lambda_K \frac{z_i}{f_i^*} & \text{if } f_i^* \neq 0, \\ 1 & \text{otherwise} \end{cases}.$$

Remark 4.5.3.5 (Uniform mass-distribution via flux penalization)**.** *The uniform mass-distribution method presented in equation* (4.16) *with interpolating parameters given by* (4.17) *can be recast in the form* (4.18) *by choosing the penalization param-*

eters to be

$$z_i = \begin{cases} \max(0, f_i^*) & \text{if } \delta_K > 0 \\ \min(0, f_i^*) & \text{if } \delta_K < 0 \\ 0 & \text{if } \delta_K = 0 \end{cases} \cdot$$

4.5.3.3 Localized mass-distribution

In this section we propose penalization parameters z_i's to redistribute the mass at sub-cell level. The idea is to penalize (modify) fluxes corresponding to degrees of freedom that created loss in mass conservation in the first place; moreover, we penalize also the neighbors of those degrees of freedom. The penalization terms are chosen based on the sign of δ_K and are designed to fulfill the assumptions in theorem 4.5.3.2. If $\delta_K = 0$ we choose $z_i = 0$ to fulfill assumption (1). Assume $\delta_K > 0$. If $f_i^* \leq 0$ we choose $z_i = 0$ to satisfy assumption (2). If $f_i^* > 0$ we choose

$$z_i = w_i + f_i^*/\lambda_K \min(0, 1 - \lambda_K w_i / f_i^*), \tag{4.19a}$$

with

$$w_i = (1-\theta) m_i |U_i^* - U_i^L| + \theta \max_{j \in N_i^*} m_j |U_j^* - U_j^H|, \tag{4.19b}$$

where $\theta \in [0, 1)$ and N_i^* is the tighter stencil described in figure 4.5. Having $|U_i^* - U_i^H|$ large indicates that clipping the i-th degree of freedom created a big contribution on losing mass conservation; therefore, the term $\max_{j \in N_i^*} m_j |U_j^* - U_j^H|$ penalizes the degrees of freedom that created loss in mass conservation. The maximum is taken to also penalize the neighbors. The term $m_i |U_i^* - U_i^L|$ is used to relax the penalization.

We use θ to tune the influence of the localized penalization. Higher θ increases the influence of the localized penalization. In all experiments in this chapter we set $\theta = 0.99$.

Note that $z_i > 0$ and since $f_i^* > 0$ assumption (4) is fulfilled. The minimum operator in the penalization term z_i is taken to assure assumption (5) is satisfied. If $\lambda_K w_i / f_i^* \leq 1$ then $z_i = w_i$; otherwise, $z_i = f_i^* / \lambda_K$. In both cases assumption (5) is fulfilled.

Now assume $\delta_K < 0$. If $f_i^* \geq 0$ we choose $z_i = 0$ to fulfill assumption (3); otherwise,

$$z_i = w_i + f_i^* / \lambda_K \min(0, 1 - \lambda_K w_i / f_i^*), \tag{4.19c}$$

where

$$w_i = -\left[(1-\theta) m_i |U_i^* - U_i^L| + \theta \max_{j \in N_i^*} m_j |U_j^* - U_j^H| \right]. \tag{4.19d}$$

It is easy to see that assumptions (4) and (5) are also satisfied.

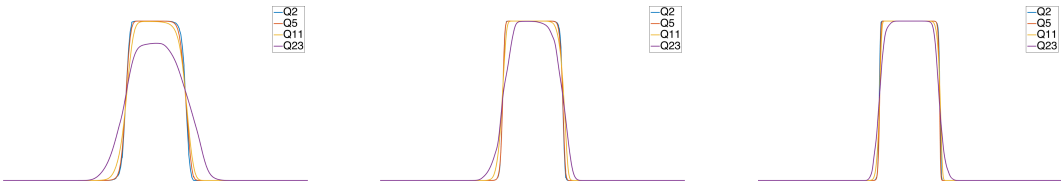
Remark 4.5.3.6 (Discontinuous, nonlinear problem per cell)**.** *To achieve mass conservation via the penalization method* (4.18) *we need* λ_K *to be given by* (4.18b)*. Since* $z_i = z_i(\lambda_K)$ *in* (4.19) *is a nonlinear function of* λ_K *we need to solve the nonlinear problem*

$$F_K(\lambda_K) := \delta_K - \lambda_K \sum_{i \in \mathcal{I}_K} z_i(\lambda_K) \equiv 0$$

to find λ_K*. This has to be done at every cell.*

In figure 4.11 we consider the 1D problem with discontinuous initial data and

show the solution with the method in this section. We consider different spaces and multiple refinements. For each refinement we adjust the number of cells to have the same number of degrees of freedom between all spaces. It is clear that more dissipation is introduced as the order of the space is increased. This is expected since the low-order method and the clipping process introduce more dissipation as the order is increased. In the next section we perform a series of convergence tests to asses the capability of the method to produce higher convergence rates as the order is increased.



(a) 192 DOFs (b) 384 DOFs (c) 768 DOFs

Figure 4.11: Element based Flux Corrected Transport with localized mass-distribution on a 1D problem with discontinuous initial condition. The method is given by (4.18), (4.19). We consider different polynomial spaces and multiple refinements. For each refinement the number of cells is adjusted to have the same number of degrees of freedom in all spaces.

4.6 Numerical examples

In this section we present numerical experiments of the three methods proposed in this chapter: the edge-based FCT method with localized stencil in §4.4, the element-based FCT with uniform mass distribution in §4.5.3.1 and the element-based FCT with localized mass distribution in §4.5.3.3. We begin by presenting the methods' converging properties on smooth and discontinuous solutions. Afterwards, we consider two benchmark problems in two dimensions.

4.6.1 Convergence tests: two dimensional smooth profile without local extrema

Consider an initial condition given by

$$u_h(x, y, t = 0) = \tanh((y - 0.5)/0.25), \tag{4.20a}$$

over $\Omega = (0, 1) \times (0, 1)$ with velocity

$$\mathbf{v} = \begin{bmatrix} \sin(\pi x)\cos(\pi y)\sin(2\pi t) \\ -\cos(\pi x)\sin(\pi y)\sin(2\pi t) \end{bmatrix} \tag{4.20b}$$

Since the velocity field is periodic and the problem is linear the exact solution at $T = 1$ coincides with the initial condition. We consider $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_3$ spaces. For this experiment we use a 4-th order with 4 stages Runge Kutta method. Tables 4.3a, 4.3b and 4.3c show the convergence rates of the edge-based FCT method with localized stencil and the element-based FCT methods with uniform and localized mass distribution respectively. We obtain the expected (optimal) convergence rates.

4.6.2 Convergence tests: one dimensional discontinuous profile

Now we consider the problem with discontinuous initial data as in §4.3. The initial condition is given by

$$u_h(\mathbf{x}, t = 0) = \begin{cases} 1, & \forall x \in (0.4, 0.6) \\ 0, & \text{otherwise} \end{cases}. \tag{4.21}$$

The domain is given by $\Omega = (0, 1) \subset \mathbb{R}$ and the velocity by $\mathbf{v} = 1$. We consider $\mathbb{Q}_2$, $\mathbb{Q}_5$ and $\mathbb{Q}_{11}$ spaces. Tables 4.4a, 4.4b and 4.4c show the convergence rates of the

Cells	$\mathbb{Q}_1$ space	rate	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_3$ space	rate
64	2.12E-03		1.89E-04		1.60E-05	
128	5.23E-04	2.01	1.72E-05	3.45	9.41E-07	4.08
256	1.29E-04	2.01	1.44E-06	3.57	5.35E-08	4.13
512	3.15E-05	2.03	1.30E-07	3.46	2.84E-09	4.23

(a) Edge-based FCT with localized stencil

Cells	$\mathbb{Q}_1$ space	rate	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_3$ space	rate
64	6.85E-03		5.30E-04		5.12E-05	
128	1.77E-03	1.94	6.18E-05	3.09	3.66E-06	3.80
256	4.18E-04	2.08	6.91E-06	3.16	2.19E-07	4.05
512	1.01E-04	2.05	7.73E-07	3.15	1.15E-08	4.25

(b) Element-based FCT with uniform mass distribution

Cells	$\mathbb{Q}_1$ space	rate	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_3$ space	rate
64	6.85E-03		5.30E-04		5.12E-05	
128	1.77E-03	1.94	6.18E-05	3.09	3.66E-06	3.80
256	4.18E-04	2.08	6.91E-06	3.16	2.19E-07	4.06
512	1.01E-04	2.05	7.73E-07	3.15	1.15E-08	4.25

(c) Element-based FCT with localized mass distribution

Table 4.3: L^1 convergence of maximum principle preserving methods using discontinuous Galerkin finite elements for a smooth solution that is monotone. We use a 4-th order with 4 stages Runge Kutta method.

edge-based FCT method with localized stencil and the element-based FCT methods with uniform and localized mass distribution respectively.

4.6.3 Convergence test: one dimensional smooth profile with local extrema

Finally we consider as initial condition

$$u_h(x, t = 0) = \cos(2\pi(x - 0.5)),$$

DOFs	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_5$ space	rate	$\mathbb{Q}_{11}$ space	rate
192	1.83E-02		5.15E-02		1.55E-01	
384	1.00E-02	0.87	3.06E-02	0.75	1.12E-01	0.46
768	5.52E-03	0.86	1.95E-02	0.65	7.92E-02	0.49
1536	3.07E-03	0.84	9.77E-03	0.99	5.55E-02	0.51

(a) Edge-based FCT with localized stencil

DOFs	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_5$ space	rate	$\mathbb{Q}_{11}$ space	rate
192	1.82E-02		2.25E-02		5.54E-02	
384	9.85E-03	0.88	1.08E-02	1.05	3.91E-02	0.50
768	5.38E-03	0.87	5.65E-03	0.93	2.95E-02	0.40
1536	2.95E-03	0.86	2.71E-03	1.05	2.16E-02	0.44

(b) Element-based FCT with uniform mass distribution

DOFs	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_5$ space	rate	$\mathbb{Q}_{11}$ space	rate
192	1.82E-02		2.25E-02		3.51E-02	
384	9.85E-03	0.88	1.07E-02	1.06	2.00E-02	0.81
768	5.38E-03	0.87	5.65E-03	0.92	1.08E-02	0.88
1536	2.95E-03	0.86	2.69E-03	1.07	6.14E-03	0.81

(c) Element-based FCT with localized mass distribution

Table 4.4: L^1 convergence of maximum principle preserving methods using discontinuous Galerkin finite elements for a discontinuous solution.

over $\Omega = (0, 1) \subset \mathbb{R}$ with velocity $\mathbf{v} = 1$. We impose periodic boundary conditions and use the initial condition as exact solution at $T = 1$. We use $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_3$ spaces. Tables 4.5a, 4.5b and 4.5c show the convergence rates of the edge-based FCT method with localized stencil and the element-based FCT methods with uniform and localized mass distribution respectively.

One can see that no better than (slightly higher than) second-order is achieved (in the L^1 norm). This issue is already discussed in [2], where the authors show that the dominating error is localized in the extremal regions, while high-order accuracy is obtained in the rest of the domain. Additional details about this problem can be found in [34], where it is shown that Total Variation Diminishing (TVD) methods

can't achieve better than second-order convergence (in the L^1 norm) around local extrema. To solve this problem, within the context of finite volumes, it is common to allow small violations on the total variation near local extrema. Popular examples are UNO [34], ENO [32,35] and WENO [58] methods. In [79,80] finite volumes and discontinuous Galerkin methods are used to obtain a solution that satisfies a strict (or global) maximum principle. To achieve high-order accuracy at local extrema, the authors reconstruct a polynomial inside cells from where the bounds are computed. Alternatively, a parameter-free smoothness indicator based on a hierarchical slope limiter for high-order DG methods may be used as regularity criterion for deactivation of FCT corrections at smooth extrema [50].

Cells	$\mathbb{Q}_1$ space	rate	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_3$ space	rate
32	2.93E-03		2.72E-03		2.18E-03	
64	6.73E-04	2.12	5.88E-04	2.20	3.95E-04	2.46
128	1.55E-04	2.12	1.19E-04	2.30	7.74E-05	2.35
256	3.57E-05	2.11	2.34E-05	2.34	1.51E-05	2.35

(a) Edge-based FCT with localized stencil

Cells	$\mathbb{Q}_1$ space	rate	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_3$ space	rate
32	2.93E-03		3.78E-03		2.00E-03	
64	6.73E-04	2.12	7.21E-04	2.39	3.82E-04	2.38
128	1.55E-04	2.12	1.32E-04	2.44	7.04E-05	2.44
256	3.57E-05	2.11	2.38E-05	2.47	1.21E-05	2.53

(b) Element-based FCT with uniform mass distribution

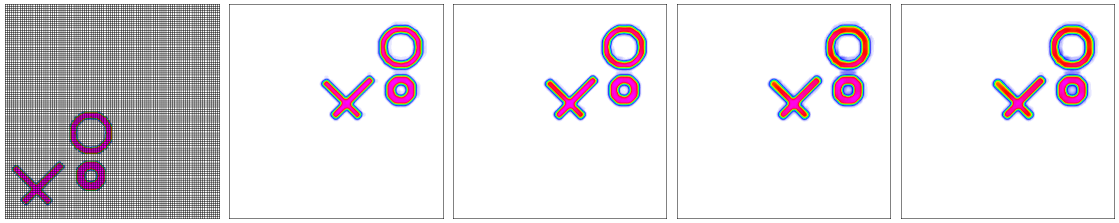
Cells	$\mathbb{Q}_1$ space	rate	$\mathbb{Q}_2$ space	rate	$\mathbb{Q}_3$ space	rate
32	2.93E-03		3.61E-03		1.99E-03	
64	6.73E-04	2.12	6.96E-04	2.43	3.82E-04	2.38
128	1.55E-04	2.12	1.29E-04	2.43	7.04E-05	2.43
256	3.57E-05	2.11	2.33E-05	2.46	1.22E-05	2.53

(c) Element-based FCT with localized mass distribution

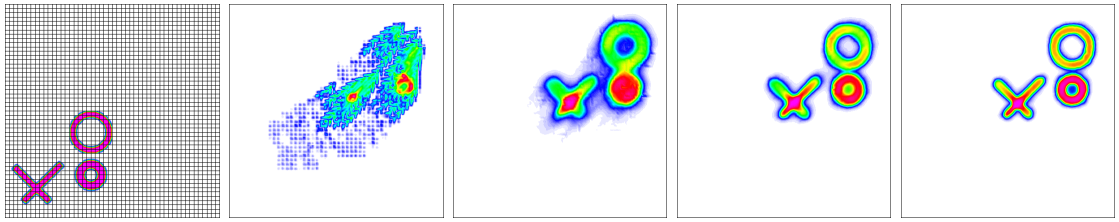
Table 4.5: L^1 convergence of maximum principle preserving methods using discontinuous Galerkin finite elements for a smooth solution with local extrema.

4.6.4 Two dimensional advection with constant velocity field

We consider $\Omega = (0, 100) \times (0, 100) \subset \mathbb{R}^2$, velocity $\mathbf{v} = (10, 10)$ and the discontinuous initial profile shown in the left panel in figure 4.12. We compute the solution using the element based FCT method with uniform mass-distribution (equations (4.16) and (4.17)) and with the localized mass-distribution (equations (4.18), (4.19)). The results are shown in figure 4.12. For comparison we also show the results of the standard FCT method revisited in §2.4 and the localized FCT method from §4.4. For all situations we use $\mathbb{Q}_2$ and $\mathbb{Q}_5$ spaces with number of cells adjusted to have 90000 degrees of freedom.



$(\min, \max) = (0, 2)$ $(-1.29\text{E-}15, 1.99)$ $(-2.30\text{E-}17, 1.97)$ $(-1.69\text{E-}18, 1.95)$ $(-8.20\text{E-}17, 1.96)$



$(\min, \max) = (0, 1)$ $(-1.68\text{E-}20, 1.32)$ $(-6.81\text{E-}15, 1.64)$ $(-5.44\text{E-}18, 1.58)$ $(-1.27\text{E-}16, 1.77)$

Figure 4.12: Two dimensional advection via different Flux Corrected Transport methods using discontinuous Galerkin finite elements. We consider **top:** $\mathbb{Q}_2$ and **bottom:** $\mathbb{Q}_5$ spaces with number of cells adjusted to have 90000 degrees of freedom in all situations. **Left:** initial condition with the mesh. **Middle-left:** solution via the standard FCT method (2.8). **Middle-middle:** solution via the localized FCT method from §4.4. **Middle-right:** solution via the element based FCT method with uniform mass-distribution (4.16), (4.17). **Right:** solution via the element based FCT method with localized mass-distribution (4.18), (4.19).

4.6.5 Two dimensional Zalesak disk

Now consider the test first proposed in [78]. The initial data is the characteristic function of a disc of radius $r = 0.15$ centered at the point $x_0 = (0.5, 0.75)$ with a thin rectangular area removed. The removed area is $\{x \in (x, y) \in \mathbb{R}^2 \mid |x - 0.5| < 0.025, y - 0.75 < 0.1125\}$. The velocity field is given by $\mathbf{v} = (-2\pi(y - 0.5), 2\pi(x - 0.5))$, which produces a rigid circular motion; therefore, the exact solution coincides with the initial data at $T = 1$. We compute the solution using the element based FCT method with uniform mass-distribution (equations (4.16) and (4.17)) and with the localized mass-distribution (equations (4.18), (4.19)). The results are shown in figure 4.13. For comparison we also show the results of the standard FCT method revisited in §2.4 and the localized FCT method from §4.4. For all situations we use $\mathbb{Q}_2$ and $\mathbb{Q}_5$ spaces with number of cells adjusted to have 90000 degrees of freedom.

4.7 Conclusions

We have presented two methods that address robustness issues with maximum principle preserving solutions of the transport equation with high-order (above $\mathbb{Q}_3$) discontinuous Galerkin spaces via the flux corrected transport. These problems are clearly depicted in figures 4.3 and 4.4. Non-physical oscillations are introduced. Both methods are based on combined effects of Bernstein polynomial basis functions, discontinuous Galerkin approximation and localized bounds.

The first method is a simple modification to the classical or standard flux corrected transport by [7] and [78]. The idea is to redefined the bounds to mimic a first-order stencil. See figure 4.5. The method highly reduces the non-physical oscillations but introduces high dissipation for higher-order spaces. We performed convergence tests and recover the expected high-order accuracy for monotone solutions (using $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_3$). When the solution is discontinuous, we observe a drop

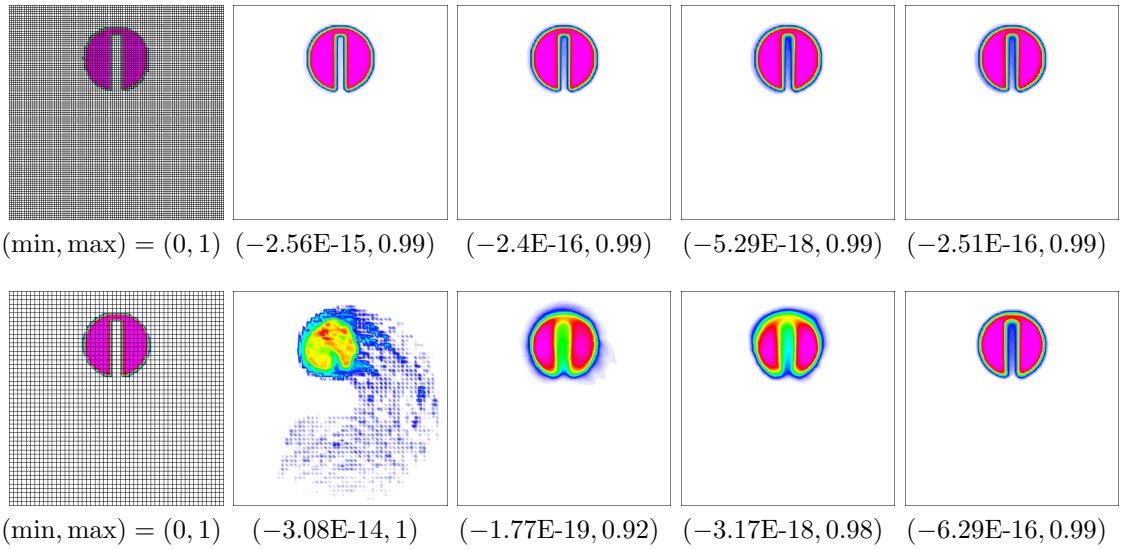


Figure 4.13: Two dimensional Zalesak disk via different Flux Corrected Transport methods using discontinuous Galerkin finite elements. We consider **top:** $\mathbb{Q}_2$ and **bottom:** $\mathbb{Q}_5$ spaces with number of cells adjusted to have 90000 degrees of freedom in all situations. **Left:** initial condition with the mesh. **Middle-left:** solution via the standard FCT method (2.8). **Middle-middle:** solution via the localized FCT method from §4.4. **Middle-right:** solution via the element based FCT method with uniform mass-distribution (4.16), (4.17). **Right:** solution via the element based FCT method with localized mass-distribution (4.18), (4.19).

in the convergence for higher-order spaces; in particular, for $\mathbb{Q}_{11}$ spaces and above. Finally, we performed a convergence study with a smooth solution with local extrema and obtained no better than second-order (in the L^1 norm) for any space. This is a known problem for methods that impose monotonicity constraints.

The second method presents a high improvement from the previous approach. It incorporates the ideas of element based flux correction and nonlinear and local mass redistribution. Opposed to the standard flux corrected transport method this approach considers two solutions on bounds: a first-order mass conservative solution and a high-order non mass conservative solution. An interpolation is performed from the low- to the high-order solution to recover mass conservation. The main

advantage of this method (opposed to the standard flux corrected transport) is the usage of a single interpolating parameter. In addition, this method gives felxibility on the process to recover mass conservation. In particular, we propose two approaches. First, we consider a linear and uniform mass distribution per cell. Later, we improve the process by localizing the mass distribution within a cell to the level of degrees of freedom. This process requires solving a nonlinear problem at every cell. We remark that the nonlinear problem is non-smooth and that since it is defined per cell the process is highly parallelizable. We performed convergence tests (for the linear and the nonlinear redistribution of mass) and recover the expected high-order accuracy for monotone solutions (using $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_3$). For the problem with discontinuous solution we obtained better results via the method with localized mass redistribution. We remark that for the higher-order spaces ($\mathbb{Q}_{11}$) we observe a clear improvement through the localized mass redistribution. When the solution is smooth with local extrema we obtained no better than second-order (in the L^1 norm) convergence.

5. ARTIFICIAL COMPRESSION WITH THE FLUX CORRECTED TRANSPORT

As explained in the introduction we are interested in applying numerical methods to solve the linear scalar conservation law to transport a level set function. We consider a smoothed Heaviside level set function that consists of constant states connected by a smooth but sharp transition as in [10, 61, 63]. To transport this level set function we assume divergence free velocity fields and solve the transport equation in conservation form. Doing this allows us to obtain mass conservative methods. We add artificial viscosity to stabilize the solution; i.e., to reduce numerical oscillations. Unfortunately, this viscosity also introduces dissipation on the level set, which leads to loss in area enclosed by the interface. For this reason it is a common practice to reinitialize the level set. A reinitialization process is meant to force the dissipated level set profile to recover its original profile. The specific reinitialization depends on the level set function itself. In the case of a smoothed Heaviside function the reinitialization is given by sharpening the interface. To do this we use artificial compression operators as in [29, 30].

In the previous chapters we used u and $\mathbf{v}$ to denote the solution and the velocity of the linear conservation law and ϕ to denote a shape function of the finite element space. This is a common notation when solving conservation laws. In this chapter and in chapter 7, however, we change the notation to a common notation when solving the level set method and for multiphase simulations. We let ϕ and $\mathbf{u}$ denote the level set function and the velocity field respectively. The shape functions of the finite element space are denoted by ψ.

We present two methods for solving the level set described in the previous paragraph. The first approach uses a high-order nonlinear viscosity and a nonlinear artificial operator based on the weak formulation of a negative Laplace's like operator. In the second approach we use an edge based artificial compression.

5.1 Non balanced artificial compression based on weak formulation of Laplace's operator

In this section we present a first approach to reinitialize the smooth Heaviside level set.

5.1.1 Formulation of the problem

We start by introducing the model in the continuous level. The idea is to consider the transport equation in conservative form. Then we add artificial viscosity. Finally, we remove some of this viscosity via a nonlinear artificial compression acting near the interface. The problem is given by

$$\partial_t \phi + \nabla \cdot \left[\mathbf{u}\phi - \mu \left(\nabla \phi - \frac{c_C(1-\phi^2)}{h||\nabla \phi||_{\ell^2}} \nabla \phi^* \right) \right] = 0, \tag{5.1a}$$

$$\phi^* - h^2 \Delta \phi^* = \phi, \tag{5.1b}$$

where μ is an artificial viscosity coefficient (to be defined), $c_C = \mathcal{O}(1)$ is a user defined constant, h is the mesh size and ϕ^* is a smooth version of ϕ. As pointed out we first add artificial viscosity to stabilize the equation. This viscosity can be linear or nonlinear depending on the coefficient μ. We consider first a linear and first-order viscosity that preserves the maximum principle. Later we improve the accuracy by using a nonlinear high-order viscosity. The last term in equation (5.1a) is responsible for removing some of the dissipation. The idea is to remove dissipation near the interface; i.e., near $\phi = 0$. In particular, observe that if $\phi = -1, 1$ the

compression is disabled. This term must change if a different level set is used. For example, if $\phi \in [0,1]$ then we use $\phi(1-\phi)$ instead of $1-\phi^2$. Using ϕ^* instead of ϕ reduces the compression of small numerical perturbations avoiding them to grow.

5.1.2 Spatial discretization

Here we describe the finite element discretization simply of the transport equation without artificial viscosity and without artificial compression. In the following sections we describe the discretization of those operators. Consider a computational mesh $\mathcal{T}_h$ and define the continuous finite dimensional space $X_h = \{\psi : \psi|_K \in \mathbb{Q}|_K, \forall K \in \mathcal{T}_h, [[\psi]] = 0\}$ where $\mathbb{Q}|_K$ is a polynomial space over the element K. We consider a Galerkin approximation; i.e., we use the space X_h for the trial and test functions. Let $\psi \in X_h$ be a shape function, multiply the transport equation (2.1) by it and integrate over the domain Ω. In addition, let $\phi_h \in X_h$ be the finite element approximation of ϕ. The problem becomes find $\phi_h \in X_h$ such that

$$\int_\Omega (\partial_t \phi_h)\psi d\mathbf{x} + \int_\Omega [\nabla \cdot (\mathbf{u}\phi_h)]\psi d\mathbf{x} = 0. \tag{5.2}$$

This equation can be recast into matrix-vector form as

$$M\frac{\partial \Phi(t)}{\partial t} + T(\Phi(t)) = 0, \tag{5.3a}$$

where $\Phi(t)$ are the degrees of freedom changing in time, M is the mass matrix with entries

$$M_{ij} = \int_\Omega \psi_i \psi_j d\mathbf{x}, \tag{5.3b}$$

and $T(\Phi(t))$ is a discretization of the transport operator acting on the solution $\Phi(t)$. We use different expressions depending on the method. The details are given in the corresponding sections (see (5.5b) and (5.16c)).

5.1.3 Time discretization

For simplicity we present the full discretization considering Forward Euler integration in time. However, we extend the results to high-order approximations in time via Strong Stability Preserving (SSP) methods [20]. Indeed, all numerical experiments, unless otherwise noted, are performed via a third-order (with three stages) Runge-Kutta SSP method. The time discretization of (5.3) via Forward Euler is given by

$$M\left(\frac{\Phi^{n+1}-\Phi^{n}}{\Delta t}\right)+T(\Phi^{n})=0, \tag{5.4}$$

where Φ^{n+1} and Φ^{n} are the degrees of freedom at time t^{n+1} and t^{n} respectively.

Remark 5.1.3.1. *(Scaling). Note that each term in equation* (5.4) *scales like*

$$(\textit{speed})\times(\textit{units of }\phi)\times\frac{|K|}{h}.$$

It is important to consider this scaling for designing some parameters with the artificial viscosity and artificial compression operators in the following sections.

5.1.4 Artificial viscosity

In this section we consider a spatial discretization of the artificial viscosity in equation (5.1a). We start in the next section with a first-order viscosity that preserves the maximum principle. Afterwards, we introduce a high-order nonlinear artificial viscosity, which enhances the accuracy properties of the solution but introduces violations

on the maximum principle. We eliminate those violations via the Flux Corrected Transport (FCT) method.

5.1.4.1 First-order viscosity

Here we consider the first-order viscosity by [23], which we revisited in §3.2.1. The method is given by

$$M^L \left(\frac{\Phi^L - \Phi^n}{\Delta t} \right) + T(\Phi^n) + D^L \Phi^n = 0, \tag{5.5a}$$

where Φ^L is the low-order solution at time t^{n+1}, M^L is the diagonal lumped mass matrix, $T(\Phi^n)$ is the column vector with entries

$$T(\Phi(t))_i = \int_\Omega \nabla \cdot (\mathbf{u}\phi_h)\psi_i d\mathbf{x} \tag{5.5b}$$

and D^L is a linear dissipative matrix with entries given by

$$D^L_{ij} = \sum_{K \in \mathcal{T}_h} \nu^L_K b_K(\psi_i, \psi_j), \tag{5.5c}$$

with

$$b_K(\psi_i, \psi_j) = \begin{cases} -\frac{|K|}{n_K - 1}, & \text{if } i \neq j, i, j \in \mathcal{I}_K \\ |K|, & \text{if } i = j, i, j \in \mathcal{I}_K \\ 0, & \text{otherwise} \end{cases}, \tag{5.5d}$$

$$\nu^L_K = \max_{\substack{i,j \in \mathcal{I}_K \\ i \neq j}} \frac{\left| \int_{S_{ij}} (\mathbf{u} \cdot \nabla \psi_j)\psi_i d\mathbf{x} \right|}{-\sum_{T \subset S_{ij}} b_T(\psi_i, \psi_j)}, \tag{5.5e}$$

where $K \in \mathcal{T}_h$ is a cell, n_K is the number of degrees of freedom in K, $\mathcal{I}_K$ is the index set of all degrees of freedom on cell K and $S_{ij} = S_i \cap S_j$ with S_i being the support

of the i-th shape function and similarly for S_j. See §3.2.1 for some properties of this method.

Remark 5.1.4.1. *(Scaling). It is easy to see that* $b_K(\psi_i, \psi_j) \sim |K|$ *and, therefore,* ν_K^L *scales like*

$$\nu_K^L \sim (speed) \times \frac{1}{h},$$

which implies that $D_{ij}^L \sim (speed) \times \frac{|K|}{h}$ *and that for any* $i \in [1, \dots, N]$ $\quad [D^L \Phi^n]_i$ *scales like*

$$[D^L \Phi^n]_i \sim (speed) \times (units\ of\ \phi) \times \frac{|K|}{h},$$

which is the correct scaling.

5.1.4.2 High-order viscosity

Now we consider a high-order artificial viscosity based on the entropy residual of the solution inspired by [25] and following [24]. The method is given by

$$M \left(\frac{\Phi^H - \Phi^n}{\Delta t} \right) + T(\Phi^n) + D^H \Phi^n = 0, \tag{5.6a}$$

where Φ^H denotes the high-order solution at time t^{n+1}, M is the consistent mass matrix, T is the column vector with entries given by (5.5b) and D^H is a high-order nonlinear artificial viscosity based on the entropy residual of the solution. The entries of the diffusive matrix D^H are

$$D_{ij}^H = \sum_{K \in \mathcal{T}_h} \nu_K^{NL} b_K(\psi_i, \psi_j), \tag{5.6b}$$

where $b_K(\cdot,\cdot)$ is the bilinear operator given by (5.5d) and

$$\nu_K^{NL} = \min\left(\nu_K^L, \frac{c_E R_K(E(\phi_h^n))}{||E(\phi_h^n) - \bar{E}(\phi_h^n)||_{L^\infty(\Omega)}}\right). \tag{5.6c}$$

Here ν_K^L is the linear viscosity given by (5.5e), $c_E = \mathcal{O}(1)$ is a user defined constant, $E(\phi_h)$ is a convex entropy function and R_K is the entropy residual. We use as entropy $E(\phi_h) = -\log(|1-\phi_h^2| + \epsilon), \epsilon = 10^{-14}$, which is specially effective for the level set we consider. The entropy residual is given by

$$R_K(E(\phi_h^n)) = \left|\left|\frac{E(\phi_h^n) - E(\phi_h^{n-1})}{\Delta t} + \frac{1}{2}\left[\mathbf{u}^n \cdot \nabla E(\phi_h^n) + \mathbf{u}^{n-1} \cdot \nabla E(\phi_h^{n-1})\right]\right|\right|_{L^\infty(K)}. \tag{5.6d}$$

Remark 5.1.4.2 (Properties of the diffusive operator D^H)**.** *The diffusive operator D^H has the same structure as D^L in §3.2.1 but with a nonlinear coefficient ν_K^{NL}. Therefore, D^H is symmetric and $\sum_i D_{ij}^H = \sum_j D_{ij}^H = 0$, see remark 3.2.1.1.*

Remark 5.1.4.3 (Scaling)**.** *Note that the term $R_K(E(\phi_h^n))$ scales like*

$$R_K(E(\phi_h^n)) \sim (\textit{speed}) \times (\textit{units of } E(\phi_h^n)) \times \frac{1}{h},$$

and therefore

$$\frac{c_E R_K(E(\phi_h^n))}{||E(\phi_h^n) - \bar{E}(\phi_h^n)||_{L^\infty(\Omega)}} \sim (\textit{speed}) \times \frac{1}{h},$$

which implies that $\nu_K^{NL} \sim (\textit{speed}) \times \frac{1}{h}$, $D_{ij}^H \sim (\textit{speed}) \times \frac{|K|}{h}$ and that for any $i \in [1,\dots,N]$ $[D^H\Phi^n]_i$ scales like

$$[D^H\Phi^n]_i \sim (\textit{speed}) \times (\textit{units of } \phi) \times \frac{|K|}{h},$$

which is the correct scaling.

5.1.5 Artificial compression

Now we incorporate the artificial compression operator. Doing so we get

$$M\left(\frac{\Phi^H - \Phi^n}{\Delta t}\right) + T(\Phi^n) + D^H\Phi^n + G\Phi^* = 0, \tag{5.7a}$$

where Φ^H denotes the high-order solution at time t^{n+1}, M is the consistent mass matrix, T is the column vector with entries given by (5.5b), D^H is the high-order dissipative operator given by (5.6b), G is a nonlinear artificial compression based on a weak formulation of the Laplace's operator and Φ^* are the degrees of freedom of ϕ^*, which is an smooth version of ϕ given by

$$\phi_h^* - h^2\Delta\phi_h^* = \phi_h. \tag{5.7b}$$

To obtain the entries of the artificial compression operator G, consider the last term in equation (5.1a), multiply it by ψ_i, integrate over the domain Ω and integrate by parts to obtain

$$-c_C\int_\Omega \frac{\mu(1-\phi_h^n)}{h||\nabla\phi_h^n||_{\ell^2}}(\nabla\phi_h^*\cdot\nabla\psi_i)d\mathbf{x}. \tag{5.7c}$$

Plug $\phi_h^* = \sum_j \Phi_j^*\psi_j$ into (5.7c) to get that the entries of the operator G are

$$G_{ij} = -c_C\left(\frac{\mu}{h}\right)\int_\Omega \frac{1-\phi_h^2}{||\nabla\phi_h^n||_{\ell^2}}(\nabla\psi_i\cdot\nabla\psi_i)d\mathbf{x}.$$

Note that G_{ij} must scale like

$$G_{ij} \sim (\text{speed}) \times \frac{|K|}{h},$$

and since the integral scales like

$$\int_\Omega \frac{1-\phi_h^2}{||\nabla \phi_h^n||_{\ell^2}} (\nabla \psi_i \cdot \nabla \psi_i) d\mathbf{x} \sim \frac{|K|}{h}$$

we need $\mu/h \sim$ (speed). We are interested in using the high-order nonlinear artificial viscosity in the previous section which scales like $\nu_K^{NL} \sim (\text{speed}) \times \frac{1}{h}$. Therefore, we propose to use μ such that

$$\frac{\mu}{h} = \nu_K^{NL} h.$$

Finally, the elements of the artificial compression matrix G are given by

$$G_{ij} = -c_C \sum_{K \in \mathcal{K}} \nu_K^{NL} h_K \int_K \frac{1-\phi_h^2}{||\nabla \phi_h||_{\ell^2}} (\nabla \psi_i \cdot \nabla \psi_j) dx, \tag{5.7d}$$

where we allow the possibility of having variable mesh size and set $h = h_K$, the mesh size of cell K.

Remark 5.1.5.1 (Properties of the artificial compression operator G)**.** *The artificial compression operator G is clearly symmetric and*

$$\begin{aligned} \sum_i G_{ij} &= \sum_i -c_C \sum_{K \in \mathcal{K}} \nu_K^{NL} h_K \int_K \frac{1-\phi_h^2}{|\nabla \phi_h|} (\nabla \psi_i \cdot \nabla \psi_j) d\mathbf{x} \\ &= -c_C \sum_{K \in \mathcal{K}} \nu_K^{NL} h_K \int_K \frac{1-\phi_h^2}{|\nabla \phi_h|} (\nabla \sum_i \psi_i \cdot \nabla \psi_j) d\mathbf{x} = 0, \end{aligned}$$

and similarly $\sum_j G_{ij} = 0$.

5.1.6 Maximum Principle Preserving solution

In this section we consider the second-order non-MPP method with artificial compression (5.7) and via the FCT methodology obtain a method that fulfills the maximum principle. To do this we need a MPP low-order method. We use the low-order method in §5.1.4.1. By subtracting the low-order method (5.5a) from the high-order method (5.7a), we obtain:

$$M^L(\Phi^H - \Phi^L) = (M^L - M)(\Phi^H - \Phi^n) + \Delta t(D^L - D^H)\Phi^n - \Delta t G\Phi^*, \tag{5.8}$$

Note that for any $i = 1, \ldots, N$ we have $\sum_j (M^L - M)_{ij} = 0$ by definition of the lumped mass matrix and $(D^L - D^H)_{ii} = -\sum_{j\neq i}(D^L - D^H)_{ij}$ since the matrices have zero column sum. Therefore,

$$\begin{aligned}[(D^L - D^H)\Phi^n]_i &= \sum_j (D^L - D^H)_{ij}\Phi_j^n = \sum_{j\neq i}(D^L - D^H)_{ij}\Phi_j^n + (D^L - D^H)_{ii}\Phi_i^n \\ &= \sum_{j\neq i}(D^L - D^H)_{ij}(\Phi_j^n - \Phi_i^n) = \sum_j (D^L - D^H)_{ij}(\Phi_j^n - \Phi_i^n),\end{aligned}$$

and since $D^L - D^H$ is symmetric, the matrix with entries $(D^L - D^H)_{ij}(\Phi_j^n - \Phi_i^n)$ is skew-symmetric. Similarly,

$$\sum_j (M^L - M)_{ij}(\Phi_j^H - \Phi_j^n) = \sum_j (M^L - M)_{ij}(\delta\Phi_j - \delta\Phi_i),$$

where $\delta\Phi := \Phi^H - \Phi^n$. Again the matrix with entries $(M^L - M)_{ij}(\delta\Phi_j - \delta\Phi_i)$ is

skew-symmetric. Finally, since G also has zero column sum we get

$$[G\Phi^*]_i = \sum_j G_{ij}\Phi_j^* = \sum_j G_{ij}(\Phi_j^* - \Phi_i^*).$$

The matrix G is symmetric which implies the matrix with entries $G_{ij}(\Phi_j^* - \Phi_i^*)$ is skew-symmetric. We introduce the so-called flux correction matrix F with entries

$$f_{ij} := (M^L - M)_{ij}(\delta\Phi_j - \delta\Phi_i) + \Delta t(D^L - D^H)_{ij}(\Phi_j^n - \Phi_i^n) - \Delta t G_{ij}(\Phi_j^* - \Phi_i^*),$$

which is skew-symmetric. Then the update for the high-order solution (5.7a) can be rewritten as

$$\Phi_i^H = \Phi_i^L + m_i^{-1}\sum_j f_{ij}, \tag{5.9}$$

to which we can apply the FCT method, see §2.4 for details. Doing this we obtain

$$\Phi_i^{n+1} = \Phi_i^L + m_i^{-1}\sum_j \alpha_{ij} f_{ij}, \tag{5.10}$$

where α_{ij} are the flux limiters.

5.2 Numerical experiments

5.2.1 One dimensional advection

Consider the initial condition given by

$$\phi_h(\mathbf{x}, t=0) = \begin{cases} 1, & \forall x \in (0.4, 0.6) \\ -1, & \text{otherwise} \end{cases}. \tag{5.11}$$

The domain is given by $\Omega = (0, 1) \subset \mathbb{R}$ and the velocity by $\mathbf{u} = 1$. We impose periodic boundary conditions; therefore, the initial condition coincides with the exact solution at $t = 1, 2, \ldots$. We consider the MPP method (5.10) and perform a convergence test using $\mathbb{Q}_1$ spaces. Table 5.1 shows the results. In figure 5.1 we show the exact solution and the solution at $t = 1$, $t = 10$ and $t = 20$ considering the low-order method (5.5), the high-order non-MPP method (5.7) and the high-order MPP method (5.10). We use $c_C = 1$ and $c_C = 0$ (i.e., with and without artificial compression). It is clear the improvement in the solution between the low- and the high-order method even without compression; i.e., using $c_C = 0$. Similarly, the numerical dissipation is clearly reduced when we incorporate artificial compression; i.e., using $c_C = 1$.

DOFs	L^1 error	rate
65	1.46E-02	
129	8.90E-03	0.71
257	4.45E-03	0.99
513	2.23E-03	1.0

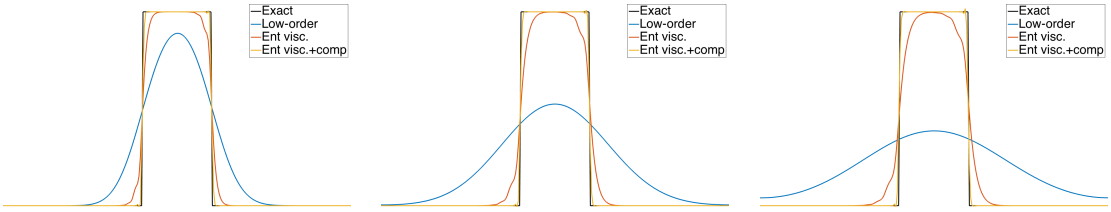
Table 5.1: L^1 convergence of non balanced artificial compression method.

5.2.2 Solid rotation: circle and ring

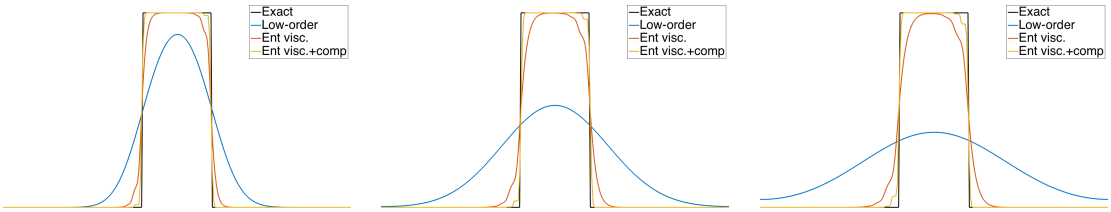
Let $r = \sqrt{(x - 0.25)^2 + (y - 0.5)^2}$, $r_0 = 0.05$, $r_1 = 0.1$ and $r_2 = 0.15$ and define the initial condition to be

$$\phi_h(\mathbf{x}, t = 0) = -\tanh\left(\frac{r - r_0}{h}\right)\tanh\left(\frac{r - r_1}{h}\right)\tanh\left(\frac{r - r_2}{h}\right), \tag{5.12a}$$

which represents the interface by a smoothed Heaviside function. The initial condition is shown in figure 5.2a. The domain is given by $\Omega = [0, 1] \times [0, 1]$ and the



(a) Non maximum principle preserving method (5.7).



(b) Maximum principle preserving method (5.10).

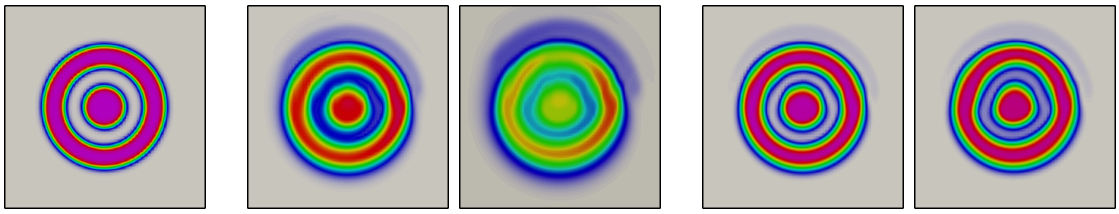
Figure 5.1: One dimensional advection problem with non balanced artificial compression. We use $\mathbb{Q}_1$ spaces and compare the low-order method (5.5) against the (a) high-order non-MPP method (5.7) and the (b) high-order MPP method (5.10) with and without compression (i.e., using $c_C = 0, 1$). The solutions are shown at times (from left to right) $t = 1, 5, 10$. The entropy coefficient is $c_E = 4$.

velocity by

$$\mathbf{u} = \begin{bmatrix} -2\pi(y - 0.5) \\ 2\pi(x - 0.5) \end{bmatrix}. \tag{5.12b}$$

The velocity field makes any initial profile to turn around the domain; therefore, after any number of complete revolutions the exact solution coincides with the initial condition. We start considering the MPP method (5.10) with $c_E = 1$ and $c_C = 0, 1$ and show in figure 5.2 surface plots at $t = 1, 4$ (one and four revolutions). In figure 5.3 we show contour plots for $\phi_h \in [-0.5, 0.5]$ at $t = 4$. For this figure we consider in the left panel the MPP method (5.10) with $c_C = 0$, in the middle panel the non-MPP method (5.7) with $c_C = 1$ and in the right panel the MPP method (5.10) with $c_C = 1$. From these experiments it is clear that the numerical dissipation is highly

reduced when artificial compression is used. It also appears that the shape of the profile is better conserved by using artificial compression and flux limiters. These experiments are performed using an structured mesh with cell size $h = 7.80 \times 10^{-3}$. Finally, we repeat the numerical experiment using a refined mesh with cell size $h = 3.90 \times 10^{-3}$. The results are shown in figure 5.4. In this case we consider $c_E = 1$ and $c_C = 1$ in figure 5.4a and $c_E = 0.5$ and $c_C = 2$ in figure 5.4b. It is clear that using larger compression constants produces sharper results. This, however, may produce undesired non-physical effects (see the right panel of figure 5.9 in §5.2.4).



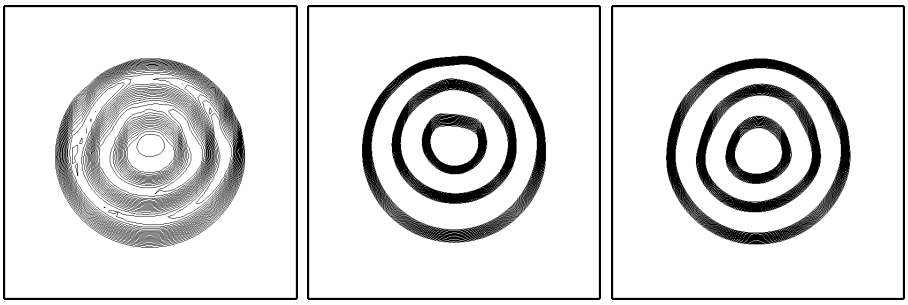
(a) Exact sol. (b) $t = 1, 4$ with $c_E = 1$, $c_C = 0$. (c) $t = 1, 4$ with $c_E = 1$, $c_C = 1$.

Figure 5.2: Surface plots of the circular rotation problem with non balanced artificial compression. We show the (a) exact solution and the solution via the MPP method (5.10) with (b) $c_C = 0$ and (c) $c_C = 1$. For each case we show the solution at (left) $t = 1$ and (right) $t = 4$. For these simulations the mesh size is $h = 7.80 \times 10^{-3}$.

5.2.3 Solid rotation: Zalesak disk

Now consider the test first proposed in [78]. The initial condition is given by

$$\phi_h(\mathbf{x}, t = 0) = \begin{cases} 1, & \text{if } (x, y) \in A \setminus B \\ -1, & \text{otherwise} \end{cases}, \tag{5.13a}$$



$(\min, \max) = (-1, 0.82)$ $(-1.01, 1.01)$ $(-1, 0.84)$

Figure 5.3: Contour plots of the circular rotation problem with non balanced artificial compression. We show contour plots for $\phi_h \in [-0.5, 0.5]$ at $t = 4$. First panel: MPP method (5.10) with $c_C = 0$. Second panel: non-MPP method (5.7) with $c_C = 1$. Third panel: MPP method (5.10) with $c_C = 1$. For these simulations the mesh size is $h = 7.80 \times 10^{-3}$.

where

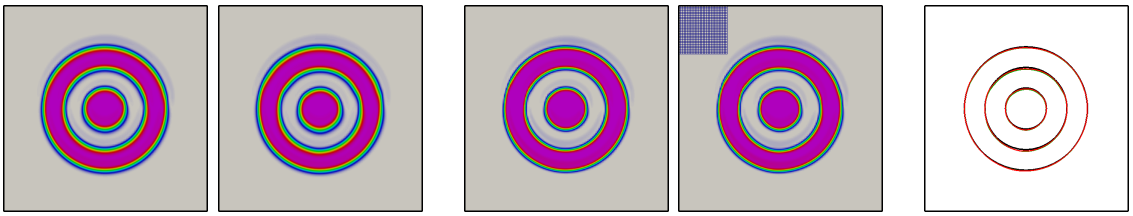
$$A = \{(x, y) \in \mathbb{R}^2 \mid \sqrt{(x - 0.5)^2 + (y - 0.75)^2} \leq 0.15\}, \tag{5.13b}$$

$$B = \{|x - 0.5| < 0.025, y - 0.75 < 0.1125\}. \tag{5.13c}$$

The velocity field is given by

$$\mathbf{u} = \begin{bmatrix} -2\pi(y - 0.5) \\ 2\pi(x - 0.5) \end{bmatrix},$$

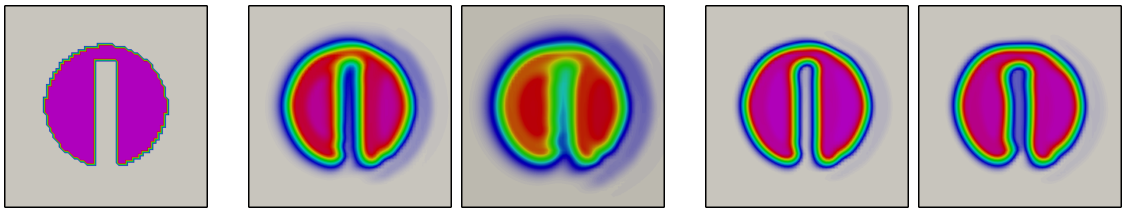
which produces a rigid circular motion so that the exact solution coincides with the initial data after any number of complete revolutions. In figure 5.5 we show the initial condition and the solution after one and four revolutions using the MPP method (5.10) with $c_E = 1$ and $c_C = 0, 1$. In addition, we show in figure 5.6 contour plots for $\phi_h \in [-0.5, 0.5]$ at times $t = 1, 2, 3, 4$. These experiments are performed using an structured mesh with cell size $h = 7.80 \times 10^{-3}$. It is clear the solution with



(a) $t = 1, 4$ with $c_E = 1$, $c_C = 1$. (b) $t = 1, 4$ with $c_E = 0.5$, $c_C = 2$. (c) $\phi_h = 0$, $t = 4$.

Figure 5.4: Refined circular rotation problem with non balanced artificial compression. We consider the MPP method (5.10) with (a) $c_E = 1$ and $c_C = 1$ and (b) $c_E = 0.5$ and $c_C = 2$ after (left) one and (right) four revolutions. In addition, we show the zero contour plots for (black) the exact solution and the solution at $t = 4$ with (green) $c_E = 1$ and $c_C = 1$ and (red) $c_E = 0.5$ and $c_C = 2$. For these simulations the mesh size is $h = 3.90 \times 10^{-3}$.

artificial compression is less dissipated. Finally, we repeat the numerical experiment using a refined mesh with cell size $h = 3.90 \times 10^{-3}$. The results are shown in figure 5.7. We use different compression constants and observe sharper results with larger compression constants.



(a) Exact sol. (b) $t = 1, 4$ with $c_E = 1$, $c_C = 0$. (c) $t = 1, 4$ with $c_E = 1$, $c_C = 1$.

Figure 5.5: Surface plots of the Zalesak disk problem with non balanced artificial compression. We show the (a) exact solution and the solution via the MPP method (5.10) with (b) $c_C = 0$ and (c) $c_C = 1$. For each case we show the solution at (left) $t = 1$ and (right) $t = 4$. For these simulations the mesh size is $h = 7.80 \times 10^{-3}$.

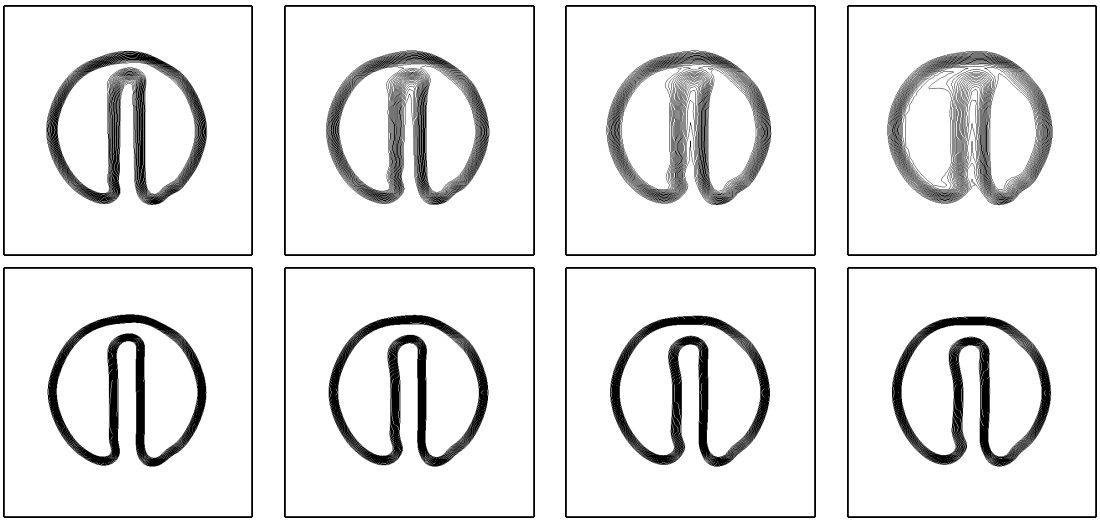


Figure 5.6: Contour plots of the Zalesak disk problem with non balanced artificial compression. We consider the MPP method (5.10) with (top) $c_E = 1$ and $c_C = 0$ and (bottom) $c_E = 1$ and $c_C = 1$ and show the solution at (from left to right) $t = 1, 2, 3, 4$. For these simulations the mesh size is $h = 7.80 \times 10^{-3}$.

5.2.4 Non-periodic vortex

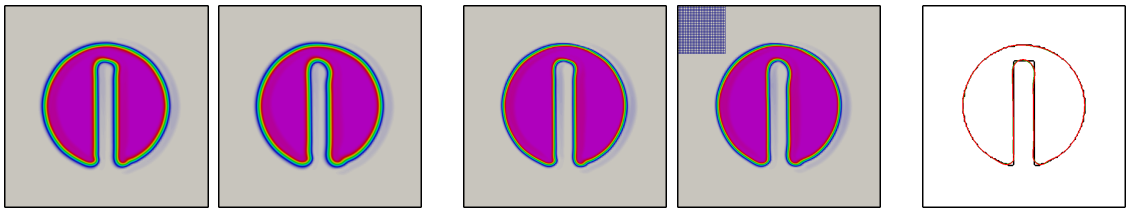
In this case we consider an initial profile and distort it via a non-periodic velocity field. The problem is given by:

$$\phi_h(\mathbf{x}, t = 0) = -\tanh\left(\frac{r - r_0}{h}\right), \tag{5.14a}$$

$$\mathbf{u} = \begin{bmatrix} -2\sin^2(\pi x)\sin(\pi y)\cos(\pi y) \\ 2\sin^2(\pi y)\sin(\pi x)\cos(\pi x) \end{bmatrix} \tag{5.14b}$$

$$\Omega = [0, 1] \times [0, 1], \tag{5.14c}$$

where $r = \sqrt{(x - 0.5)^2 + (y - 0.75)^2}$ and $r_0 = 0.15$. The velocity profile distorts the initial profile into thin regions where numerical dissipation can fade them creating the zero contour plot $\{\phi_h = 0\}$ to be lost. In these cases artificial compression



(a) $t = 1, 4$ with $c_E = 1$, $c_C = 1$. (b) $t = 1, 4$ with $c_E = 0.5$, $c_C = 2$. (c) $\phi_h = 0$, $t = 4$.

Figure 5.7: Refined Zalesak disk problem with non balanced artificial compression. We consider the MPP method (5.10) with (a) $c_E = 1$ and $c_C = 1$ and (b) $c_E = 0.5$ and $c_C = 2$ after (left) one and (right) four revolutions. In addition, we show the zero contour plots for (black) the exact solution and the solution at $t = 4$ with (green) $c_E = 1$ and $c_C = 1$ and (red) $c_E = 0.5$ and $c_C = 2$. For these simulations the mesh size is $h = 3.90 \times 10^{-3}$.

helps preventing this problem. In figure 5.8 we show the zero contour plot at $t = 0, 1, 2, 3$ and 4 considering the MPP method (5.10) with $c_E = 1$ and $c_C = 0, 1$. These experiments are performed using an structured mesh with cell size $h = 7.80 \times 10^{-3}$. One can appreciate that using artificial compression reduces numerical dissipation that helps preserving the zero contour plot. Finally, in figure 5.9 we repeat this numerical experiment with a refined mesh. The mesh size is given by 3.90×10^{-3}. We use in the left panel $c_E = 1$ and $c_C = 1$, in the middle panel $c_E = 0.5$ and $c_C = 2$ and in the right panel $c_E = 5$ and $c_C = 20$. We remark that using larger compression constants produce sharper regions; nevertheless, this might produce non-physical effects. In the right panel of figure 5.9 the zero level set in the thin regions detaches. For this reason we advise caution with the use of large compression constants.

5.3 Self balanced artificial compression based on an edge-based dissipative operator

Now we present a second approach to reinitialize the smooth Heaviside level set. We remark that we perform some one dimensional experiments and convergence tests on

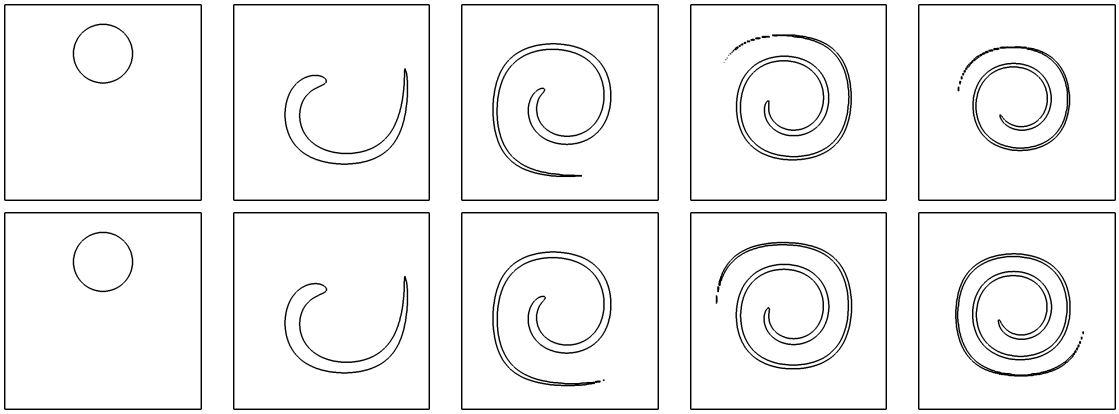


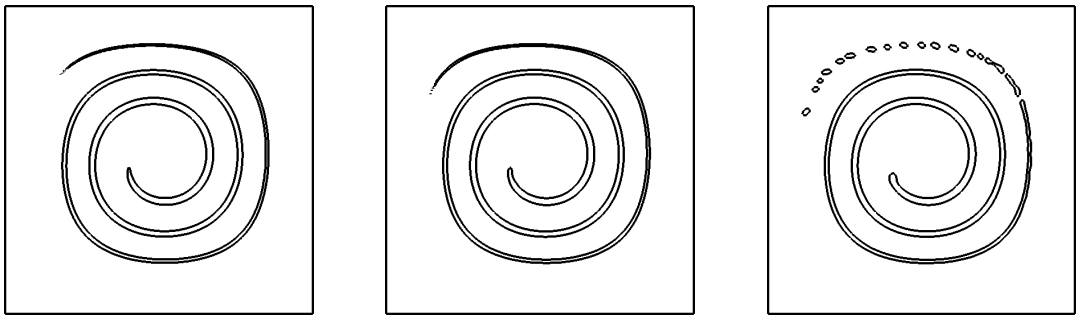
Figure 5.8: Non-periodic vortex with non balanced artificial compression. We consider the MPP method (5.10) with (top) $c_E = 1$ and $c_C = 0$ and (bottom) $c_E = 1$ and $c_C = 1$ and show the zero contour plot at (from left to right) $t = 0, 1, 2, 3, 4$. For these simulations the mesh size is $h = 7.80 \times 10^{-3}$.

$\mathbb{Q}_2$ spaces. We use $\mathbb{Q}_1$ spaces for all the numerical experiments in two dimensions.

5.3.1 Formulation of the problem

Here we introduce the model in the continuous level. The idea, as with the first approach in §5.1, is to consider the transport equation in conservative form. Then we add artificial viscosity and remove some of it via nonlinear artificial compression acting near the interface. In §5.1 the artificial compression can be stronger that the artificial viscosity, which might over compress the solution. This is controlled by the viscosity and compression coefficients c_E and c_C respectively. This is part of the reason for using the smoothed solution ϕ_h^* instead of ϕ_h in (5.1) (to reduce over compression). Now we use a model that overcomes this problem. This is given by

$$\partial_t \phi + \nabla \cdot \left[\mathbf{u}\phi - \mu \left(1 - \frac{c_C (1-\phi^2)^+}{h||\nabla \phi||_{\ell^2}} \right)^+ \nabla \phi \right] = 0, \tag{5.15}$$



$c_E = 1$, $c_C = 1$ $c_E = 0.5$, $c_C = 2$ $c_E = 5$, $c_C = 20$

Figure 5.9: Refined non-periodic vortex with non balanced artificial compression. We consider the MPP method (5.10) with (left) $c_E = 1$ and $c_C = 1$, (middle) $c_E = 0.5$ and $c_C = 2$ and (right) $c_E = 5$ and $c_C = 20$ and show the zero contour plot at $t = 4$. For these simulations the mesh size is $h = 3.90 \times 10^{-3}$.

where $c_C = \mathcal{O}(1)$ is a user defined constant and μ is an artificial viscosity coefficient (to be defined). In this case we can see the compression as a modifier of the artificial viscosity coefficient. By taking the $(\cdot)^+$ operator, we never over compress the solution; i.e., the combination of artificial viscosity and compression always remains as viscosity.

5.3.2 Spatial discretization

We use the finite element space as in §5.1.2. We start considering just the transport equation without artificial viscosity and without artificial compression. Since we consider the first-order viscosity by [26] (see §3.2.2) we follow the authors to define the discretization of the transport operator. Doing so the discretization of the transport equation becomes

$$M\frac{\partial \Phi(t)}{\partial t} + T(\Phi(t)) = 0, \tag{5.16a}$$

where M is the mass matrix with entries

$$M_{ij} = \int_\Omega \psi_i \psi_j d\mathbf{x}, \tag{5.16b}$$

and $T(\Phi(t)$ is the column vector with entries

$$T(\Phi(t))_i = \sum_j (\mathbf{u}\phi_h)_j \cdot \mathbf{c}_{ij}, \tag{5.16c}$$

where $\mathbf{c}_{ij} = \int_\Omega \nabla\psi_j \psi_i d\mathbf{x}$ and $(\mathbf{u}\phi_h)_j, j = 1, ..., N$ are the degrees of freedom of the projection of $\mathbf{u}\phi_h$ onto the finite element space; i.e., they are given by

$$\sum_j (\mathbf{u}\phi_h)_j \int_\Omega \psi_i \psi_j d\mathbf{x} = \int_\Omega (\mathbf{u}\phi_h)\psi_i d\mathbf{x}. \tag{5.16d}$$

5.3.3 Time discretization

Just as in §5.1.3 we present the full discretization considering Forward Euler integration in time. However, all experiments are obtained using a third-order with three stages Strong Stability Preserving Runge Kutta method, see [20]. The time discretization of (5.16) via Forward Euler is given by

$$M\left(\frac{\Phi^{n+1} - \Phi^n}{\Delta t}\right) + T(\Phi^n) = 0, \tag{5.17}$$

where Φ^{n+1} and Φ^n are the degrees of freedom at time t^{n+1} and t^n respectively.

Remark 5.3.3.1. *(Scaling) Note that each term in* (5.17) *scales like*

$$(\textit{speed}) \times (\textit{units of } \phi) \times \frac{|K|}{h}.$$

It is important consider this scaling for designing some parameters with the artificial viscosity and the artificial compression operators in the following sections.

5.3.4 Artificial viscosity

In this section we consider a spatial discretization of the artificial viscosity in equation (5.15). We start in the next section with a first-order method that preserves the maximum principle. Afterwards, we introduce a high-order nonlinear artificial viscosity. This viscosity enhances the accuracy properties of the solution but introduces violations on the maximum principle, which we later eliminate via the Flux Corrected Transport method.

5.3.4.1 First-order viscosity

As explained in the beginning of this section we are interested in using $\mathbb{Q}_1$ and $\mathbb{Q}_2$ spaces. For this reason we need a first-order maximum principle preserving method that is suitable for high-order spaces. Based on the conclusions of chapter 3 (see §3.2.2) we decide to use the low-order method by [26]. This method is given by

$$M^L \left(\frac{\Phi^L - \Phi^n}{\Delta t} \right) + T(\Phi^n) + D^L \Phi^n = 0, \tag{5.18a}$$

where Φ^L denotes the low-order solution at time t^{n+1}, M^L is the lumped mass matrix, $T(\Phi^n)$ is the column vector with entries given by (5.16c) and D^L is a dissipative matrix with entries given by

$$D^L_{ij} = -\max(|\mathbf{u}_i \cdot \mathbf{c}_{ij}|, |\mathbf{u}_j \cdot \mathbf{c}_{ji}|), \quad \forall i \neq j, \tag{5.18b}$$

and $D_{ii}^L = -\sum_{j\neq i} D_{ij}^L$. Here

$$\mathbf{c}_{ij} = \int_\Omega \psi_i \nabla \psi_j d\mathbf{x}. \tag{5.18c}$$

See §3.2.2 for some properties of this method.

Remark 5.3.4.1. *(Scaling) Note that* $\mathbf{c}_{ij} \sim \frac{|K|}{h}$*, which implies that* $D_{ij}^L \sim$ *(speed)* $\times$ $\frac{|K|}{h}$ *and that for any* $i \in [1, \ldots, N]$ $\quad [D^L\Phi^n]_i$ *scales like*

$$[D^L\Phi^n]_i \sim (\textit{speed}) \times (\textit{units of } \phi) \times \frac{|K|}{h},$$

which is the correct scaling.

5.3.4.2 High-order viscosity

We add a high-order nonlinear stabilization in the spirit of [25] to get

$$M\left(\frac{\Phi^H - \Phi^n}{\Delta t}\right) + T(\Phi^n) + D^H\Phi^n = 0, \tag{5.19a}$$

where Φ^H denotes the high-order solution at time t^{n+1}, M is the consistent mass matrix, $T(\Phi^n)$ is the column vector with entries given by (5.16c) and D^H is a nonlinear high-order artificial viscosity with entries given by

$$D_{ij}^H = -\min\left(|D_{ij}^L|, \frac{c_E R_{ij}}{||E(\phi_h^n) - \bar{E}(\phi_h^n)||_{L^\infty(\Omega)}}\right), \quad \forall i \neq j, \tag{5.19b}$$

and $D_{ii}^H = -\sum_{j\neq i} D_{ij}^H$. Here D_{ij}^L are the entries of the low-order dissipative operator with entries given by (5.18b) and R_{ij} is the entropy residual defined as follows:

$$R_{ij} = \frac{\left|\int_\Omega \frac{E(\phi_h^n) - E(\phi_h^{n-1})}{\Delta t} + \frac{1}{2}\left(\mathbf{u}^n \cdot \nabla E(\phi_h^n) + \mathbf{u}^{n-1} \cdot \nabla E(\phi_h^{n-1})\right)\psi_i\psi_j d\mathbf{x}\right|}{\frac{1}{|S_i \cap S_j|}\int_\Omega \psi_i\psi_j d\mathbf{x}}, \tag{5.19c}$$

where S_i is the support of the i-th shape function and similarly for S_j, $E(\phi) = -\log(|1-\phi_h^2|+1E-14)+\epsilon), \epsilon = 1\times 10^{-14}$ is a convex entropy function. Note that since we use positive basis functions given by Bernstein polynomials $\int_\Omega \psi_i\psi_j d\mathbf{x} > 0$ for polynomial spaces of any order.

Remark 5.3.4.2 (Properties of the diffusive operator D^H)**.** *The diffusive operator D^H is symmetric, has non-positive off-diagonal entries and $\sum_j D_{ij}^H = \sum_i D_{ij}^H$ (i.e., has zero column/row sum).*

Remark 5.3.4.3 (Scaling)**.** *Note that $R_{ij} \sim$ (speed) $\times$ (units of $E(\phi_h^n)$) $\times \frac{|K|}{h}$, which implies that $D_{ij}^H \sim$ (speed) $\times \frac{|K|}{h}$ and that for any $i \in [1,\dots]$ $[D^H\Phi^n]_i$ scales like*

$$[D^H\Phi^n]_i \sim (\textit{speed}) \times (\textit{units of } \phi) \times \frac{|K|}{h},$$

which is the correct scaling.

5.3.5 Artificial compression

We now introduce the artificial compression. Doing this leads to

$$M\left(\frac{\Phi^C-\Phi^n}{\Delta t}\right) + T(\Phi^n) + D^C\Phi^n = 0, \tag{5.20a}$$

where Φ^C is the artificially compressed solution at time t^{n+1}, M is the consistent mass matrix, $T(\Phi^n)$ is the column vector with entries given by (5.16c) and D^C is a high-order viscosity whose entries are given by

$$D_{ij}^C = D_{ij}^H\left[1-\frac{c_C[1-(\Phi_{ij}^n)^2]^+}{|\Phi_i-\Phi_j|}\right]^+, \quad \text{if } i\neq j \tag{5.20b}$$

and $D_{ii}^C = -\sum_{j\neq i} D_{ij}^C$. Here $\Phi_{ij} := \frac{1}{2}(\Phi_i+\Phi_j)$. Note that D^C inherits the same structure as D^H. Moreover, the artificial compression is simply a nonlinear modifi-

cation of the strength of the viscosity operator D^H. Note that for this to be true the compression term must be dimensionless, which is clearly true. Taking the positive part in (5.20b) assures D^C_{ij} has always the correct sign to be a dissipative operator. This is one of the main advantages of this approach in comparison to the method introduced in §5.1.

5.3.6 Maximum Principle Preserving solution

The last step is to obtained a maximum principle preserving solution via the Flux Corrected Transport method. We do this through two approaches.

5.3.6.1 Low-order to artificially compressed solution via Flux Corrected Transport

The first approach consists on applying directly the FCT method to (5.20). Firstly, we need a MPP low-order solution. We use the method in §5.3.4.1. Then, we require a high-order non-MPP method. We use the method in §5.3.5. By subtracting the low-order method (5.18a) from the high-order method (5.20a) we obtain

$$M^L(\Phi^C - \Phi^L) = (M^L - M)(\Phi^C - \Phi^n) + \Delta t(D^L - D^C)\Phi^n.$$

Note that for any $i = 1, \dots, N$ we have $\sum_j (M^L - M)_{ij} = 0$ by definition of the lumped mass matrix and $(D^L - D^C)_{ii} = -\sum_{j\neq i}(D^L - D^C)_{ij}$, by definition of the diagonal terms in D^L and D^C. Therefore,

$$\begin{aligned}[(D^L - D^C)\Phi^n]_i &= \sum_j (D^L - D^C)_{ij}\Phi^n_j = \sum_{j\neq i}(D^L - D^C)_{ij}\Phi^n_j + (D^L - D^C)_{ii}\Phi^n_i \\ &= \sum_{j\neq i}(D^L - D^C)_{ij}(\Phi^n_j - \Phi^n_i) = \sum_j (D^L - D^C)_{ij}(\Phi^n_j - \Phi^n_i),\end{aligned}$$

and since $D^L - D^C$ is symmetric, the matrix with entries $(D^L - D^C)_{ij}(\Phi_j^n - \Phi_i^n)$ is skew-symmetric. Similarly,

$$\sum_j (M^L - M)_{ij}(\Phi_j^C - \Phi_j^n) = \sum_j (M^L - M)_{ij}(\delta\Phi_j - \delta\Phi_i),$$

where $\delta\Phi := \Phi^C - \Phi^n$. Again the matrix with entries $(M^L - M)_{ij}(\delta\Phi_j - \delta\Phi_i)$ is skew-symmetric. We introduce the so-called flux correction matrix F with entries

$$f_{ij} := (M^L - M)_{ij}(\delta\Phi_j - \delta\Phi_i) + \Delta t(D^L - D^C)_{ij}(\Phi_j^n - \Phi_i^n),$$

which is skew-symmetric. Then the update for the high-order solution (5.20a) can be rewritten as

$$\Phi_i^C = \Phi_i^L + m_i^{-1} \sum_j f_{ij},$$

to which we can apply the FCT method, see §2.4 for details. Doing this we obtain

$$\Phi_i^{n+1} = \Phi_i^L + m_i^{-1} \sum_j \alpha_{ij} f_{ij}, \tag{5.21}$$

where α_{ij} are the flux limiters.

5.3.6.2 High-order to artificially compressed solution via Flux Corrected Transport

Here we present a second approach. First we obtain a high-order maximum principle preserving solution without the artificial compression. Afterwards, we use this solution as "low-order" maximum principle preserving solution and combine it with the artificially compressed solution via the FCT method. We recall the methods in

§5.3.4.1, §5.3.4.2 and §5.3.5.

$$M^L\left(\Phi^L-\Phi^n\right)+\Delta t\left[T(\Phi^n)+D^L\Phi^n\right]=0, \tag{5.22a}$$

$$M\left(\Phi^H-\Phi^n\right)+\Delta t\left[T(\Phi^n)+D^H\Phi^n\right]=0, \tag{5.22b}$$

$$M\left(\Phi^C-\Phi^n\right)+\Delta t\left[T(\Phi^n)+D^C\Phi^n\right]=0, \tag{5.22c}$$

where Φ^L, Φ^H and Φ^C denote the low-order, high-order and the artificially compressed solution respectively. By combining (5.22a) with (5.22b) we obtain

$$m_i(\Phi_i^H-\Phi_i^L)=\sum_j f_{ij}^H, \tag{5.23}$$

where

$$f_{ij}^H=(M^L-M)_{ij}(\delta\Phi_j^H-\delta\Phi_i^H)+\Delta t(D^L-D^H)_{ij}(\Phi_j^n-\Phi_i^n),$$
$$\delta\Phi^H=\Phi^H-\Phi^n.$$

Applying the FCT method to (5.23) yields

$$m_i(\tilde{\Phi}_i^H-\Phi_i^L)=\sum_j \alpha_{ij}^H f_{ij}^H, \tag{5.24}$$

where $\tilde{\Phi}^H$ is the limited high-order solution and α_{ij}^H are the corresponding flux limiters. Now combine equation (5.24) with (5.22c) to get

$$m_i(\Phi_i^C-\tilde{\Phi}_i^H)=\sum_j f_{ij}^C-\alpha_{ij}^H f_{ij}^H, \tag{5.25}$$

where

$$f_{ij}^C = (M^L - M)_{ij}(\delta\Phi_j^C - \delta\Phi_i^C) + \Delta t(D^L - D^C)_{ij}(\Phi_j^n - \Phi_i^n),$$
$$\delta\Phi^C = \Phi^C - \Phi^n.$$

Applying the FCT method to (5.25) yields

$$m_i(\Phi_i^{n+1} - \tilde{\Phi}_i^H) = \sum_j \alpha_{ij}^C \left(f_{ij}^C - \alpha_{ij}^H f_{ij}^H\right), \tag{5.26}$$

where Φ_i^{n+1} is the maximum principle preserving artificially compressed solution at time t^{n+1} and α_{ij}^C are the flux limiters of the flux $f_{ij}^C - \alpha_{ij}^H f_{ij}^H$. Finally, by plugging in (5.24) into (5.26) we obtain

$$m_i(\Phi_i^{n+1} - \Phi_i^L) = \sum_j \alpha_{ij}^C f_{ij}^C + (1 - \alpha_{ij}^C)\alpha_{ij}^H f_{ij}^H. \tag{5.27}$$

Since the limiters $0 \leq \alpha_{ij}^C \leq 1$ (see (2.9)) we can interpret the right hand side of (5.27) as a convex combination of two fluxes: a non-limited flux f_{ij}^C and a limited flux $\alpha_{ij}^H f_{ij}^H$.

5.4 Numerical experiments

5.4.1 One dimensional advection

Consider the initial condition given by

$$\phi_h(\mathbf{x}, t = 0) = \begin{cases} 1, & \forall x \in (0.4, 0.6) \\ -1, & \text{otherwise} \end{cases}. \tag{5.28}$$

The domain is given by $\Omega = (0, 1) \subset \mathbb{R}$ and the velocity by $\mathbf{u} = 1$. We impose periodic boundary conditions so that the initial condition coincides with the exact solution at $t = 1, 2, \ldots$. We consider the MPP methods (5.21) and (5.27) and perform a convergence test using $\mathbb{Q}_1$ and $\mathbb{Q}_2$ spaces. Table 5.2 shows the results. For each row we adjust the number of cells to have the same number of degrees of freedom. Better convergence rates and eventually smaller errors are obtained with $\mathbb{Q}_2$ spaces. It is important to emphasize that although the method (5.27) gives better results than (5.21) the improvement is negligible. This is also observed in two dimensional tests (see next section).

DOFs	$\mathbb{Q}_1$ space	rate	$\mathbb{Q}_2$ space	rate
65	8.05E-02		9.31E-02	
129	4.90E-02	0.71	4.82E-02	0.95
257	2.98E-02	0.71	2.67E-02	0.85
513	1.80E-02	0.72	1.47E-02	0.85

(a) MPP artificially compressed solution via (5.21).

DOFs	$\mathbb{Q}_1$ space	rate	$\mathbb{Q}_2$ space	rate
65	8.01E-02		9.18E-02	
129	4.86E-02	0.71	4.74E-02	0.95
257	2.96E-02	0.71	2.64E-02	0.84
513	1.78E-03	0.73	1.46E-02	0.85

(b) MPP artificially compressed solution via (5.27).

Table 5.2: L^1 convergence of self balanced artificial compression method.

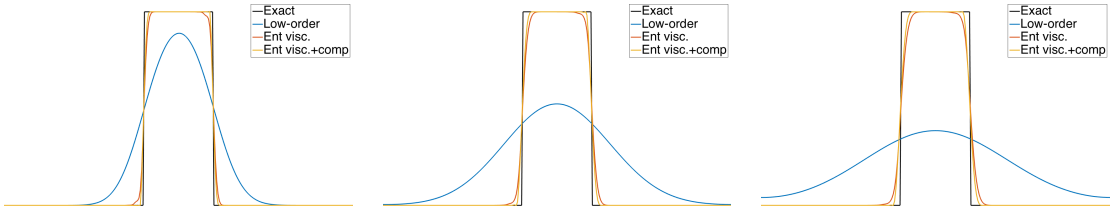
We now explore the qualitative behavior of the method using the same one-dimensional problem. In figures 5.10a and 5.10b we use $\mathbb{Q}_1$ and $\mathbb{Q}_2$ spaces and compare the low- and the high-order non-MPP method with entropy viscosity with and without compression. For now we don't use flux limiters; i.e., we use method

(5.20) with $c_C = 0, 1$. It is clear the improvement from the low- to the high-order solution. Similarly, it is easy to observe the reduction in numerical dissipation when we use artificial compression. The results are qualitatively similar with $\mathbb{Q}_1$ and $\mathbb{Q}_2$ spaces. The next step is to use flux limiters via the FCT methodology; i.e., to use method (5.21) or (5.27) (we use the latter). We do this and show the results in figures 5.10c and 5.10d. First observe figure 5.10c where $\mathbb{Q}_1$ spaces are used. In the left we don't use artificial compression and compare the solution via the high-order entropy viscosity method with and without limitation. The solutions are close together. The limitation process doesn't change the solution drastically (for this example). The reason for this is that the solution with entropy viscosity is close to be maximum principle preserving; therefore, minor limitation is needed. Now consider the figure on the right in 5.10c. Here we compare the solution with artificial compression with and without limitation. It is clear the limitation changes the solution considerably. In this experiment, using the FCT eliminates most of the improvement in the solution in order to preserve the maximum principle. In the next sections we consider other problems where the improvement by using artificial compression and limitation via the FCT is more evident. Finally we remark that (for this experiment) the improvement by using artificial compression within the FCT methodology is more evident with $\mathbb{Q}_2$ spaces. See figure 5.10d.

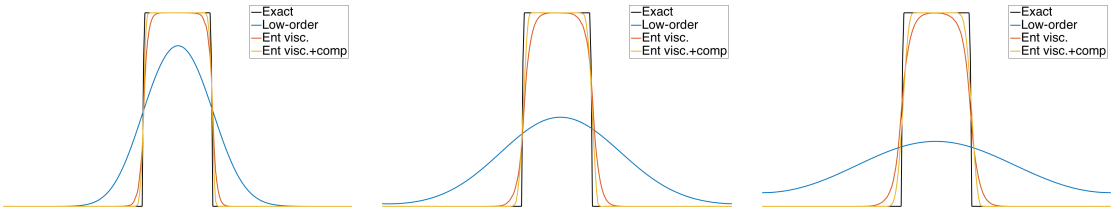
5.4.2 Solid rotation: circle and ring

Let $r = \sqrt{(x-0.25)^2 + (y-0.5)^2}$, $r_0 = 0.05$, $r_1 = 0.1, r_2 = 0.15$ and consider the initial condition

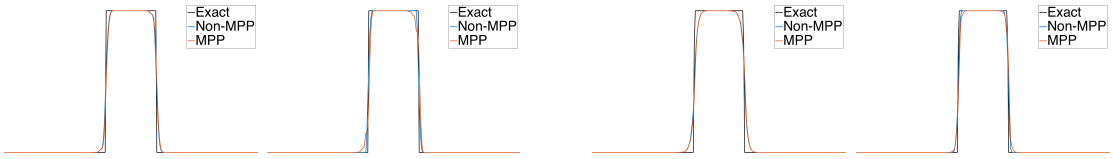
$$\phi_h(\mathbf{x}, t=0) = -\tanh\left(\frac{r-r_0}{h}\right)\tanh\left(\frac{r-r_1}{h}\right)\tanh\left(\frac{r-r_2}{h}\right), \tag{5.29a}$$



(a) Low-order and non-MPP method (5.20) with $c_E = 1$ and $c_C = 0, 1$ using $\mathbb{Q}_1$ spaces.



(b) Low-order and non-MPP method (5.20) with $c_E = 1$ and $c_C = 0, 1$ using $\mathbb{Q}_2$ spaces.



(c) $\mathbb{Q}_1$ space (d) $\mathbb{Q}_2$ space

Figure 5.10: One dimensional advection problem with self balanced artificial compression. In (a) and (b) we use $\mathbb{Q}_1$ and $\mathbb{Q}_2$ spaces respectively and compare the low-order and the high-order solutions with and without compression (i.e., using $c_C = 0, 1$) for (from left to right) $t = 1, 5, 10$. In (c) and (d) we use $\mathbb{Q}_1$ and $\mathbb{Q}_2$ spaces respectively. For each case we compare (left) the high-order method using $c_E = 1$ and $c_C = 0$ with and without limitation and (right) the high-order method using $c_E = 1$ and $c_C = 1$ with and without limitation.

which represents the interface by a smoothed Heaviside function. The initial condition is shown in figure 5.11a. The domain is given by $\Omega = [0, 1] \times [0, 1]$ and the velocity by

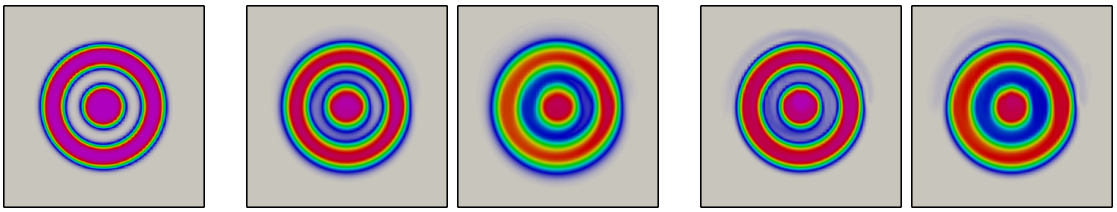
$$\mathbf{u} = \begin{bmatrix} -2\pi(y - 0.5) \\ 2\pi(x - 0.5) \end{bmatrix}. \tag{5.29b}$$

The velocity field makes any initial profile to turn around the domain; therefore, after one rotation the exact solution coincides with the initial condition. We considered the MPP method (5.21) with $c_E = 1$ and $c_C = 0, 1$ and use an structured mesh with cell size $h = 7.80 \times 10^{-3}$. In figure 5.11 we show surface plots of the (zoomed) initial condition and the solution at times $t = 1$ and 4. In figure 5.12 we show contour plots for $\phi_h \in [-0.5, 0.5]$ at $t = 4$ using different methods. In the first panel we use the MPP method (5.21) with $c_C = 0$. In the second panel we use the non-MPP method (5.20) with $c_C = 1$. The third panel corresponds to the MPP method (5.21) with $c_C = 1$. Finally, the fourth panel shows the solution of the MPP method (5.27) with $c_C = 1$. Firstly, we observe that the solution is less dissipated by using artificial compression (all panels except the first one). Secondly, we remark that, as expected, not using limiters yields sharper solutions (second panel). Finally, it is important to note (as in §5.4.1 with the one dimensional experiment) that both MPP methods (5.21) and (5.27) produce similar results (third and fourth panel). Finally, we repeat the numerical experiment using a refined mesh with cell size $h = 3.90 \times 10^{-3}$. The results are shown in figure 5.13. In this case we consider $c_E = 1$ and $c_C = 1$ in figure 5.13a and $c_E = 0.5$ and $c_C = 2$ in figure 5.13b. We remark that using larger compression constants produces no significant changes in the solution. This is due to the $(\cdot)^+$ operator in (5.15). This remark is more evident in figure 5.13c where we compare the zero contour plots.

5.4.3 Solid rotation: Zalesak disk

Now consider the test first proposed in [78]. The initial condition is given by

$$\phi_h(\mathbf{x}, t = 0) = \begin{cases} 1, & \text{if } (x, y) \in A \setminus B \\ -1, & \text{otherwise} \end{cases}, \tag{5.30a}$$



(a) Exact sol. (b) $t = 1, 4$ with $c_E = 1, c_C = 0$ (c) $t = 1, 4$ with $c_E = 1, c_C = 1$

Figure 5.11: Surface plots of the circular rotation problem with self balanced artificial compression. We show the (a) initial exact solution and the solution via the MPP method (5.21) with (b) $c_C = 0$ and (c) $c_C = 1$. For each case we show the solution at (left) $t = 1$ and (right) $t = 4$. For these simulations the mesh size is $h = 7.80 \times 10^{-3}$.

where

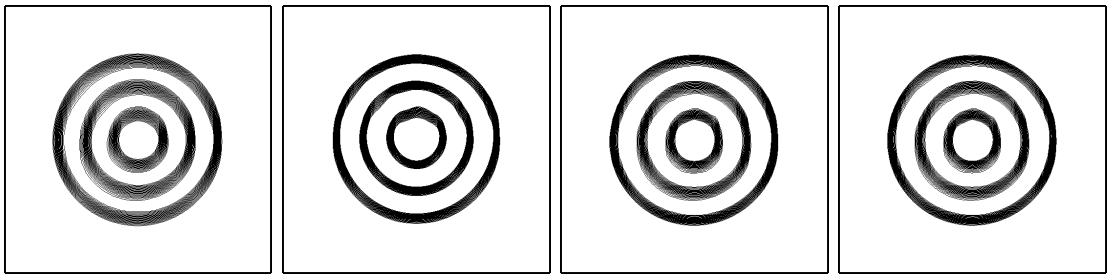
$$A = \{(x, y) \in \mathbb{R}^2 \mid \sqrt{(x - 0.5)^2 + (y - 0.75)^2} \leq 0.15\}, \tag{5.30b}$$

$$B = \{|x - 0.5| < 0.025, y - 0.75 < 0.1125\}. \tag{5.30c}$$

The velocity field is given by

$$\mathbf{u} = \begin{bmatrix} -2\pi(y - 0.5) \\ 2\pi(x - 0.5) \end{bmatrix},$$

which produces a rigid circular motion so that the exact solution coincides with the initial data after any number of complete revolutions. In figure 5.14 we show the initial condition and the solution after one and four revolutions using the MPP method (5.21) with $c_E = 1$ and $c_C = 0, 1$. In addition, we show in figure 5.15 contour plots for $\phi_h \in [-0.5, 0.5]$ at times $t = 1, 2, 3, 4$. These experiments are performed using an structured mesh with cell size $h = 7.80 \times 10^{-3}$. It is clear the solution with artificial compression is less dissipated. Finally, we repeat the numerical experiment



$(\min, \max) = (-1, 0.82)$ $(-1.01, 1.01)$ $(-1, 0.84)$ $(-1, 0.85)$

Figure 5.12: Contour plots of the circular rotation problem with self balanced artificial compression. We show contour plots for $\phi_h \in [-0.5, 0.5]$ at $t = 4$. First panel: MPP method (5.21) with $c_C = 0$. Second panel: non-MPP method (5.20) with $c_C = 1$. Third panel: MPP method (5.21) with $c_C = 1$. Fourth panel: MPP method (5.27) with $c_C = 1$. For these simulations the mesh size is $h = 7.80 \times 10^{-3}$.

using a refined mesh with cell size $h = 3.90 \times 10^{-3}$. The results are shown in figure 5.16. Again, we observe no significant difference as we increase the compression constant.

5.4.4 Non-periodic vortex

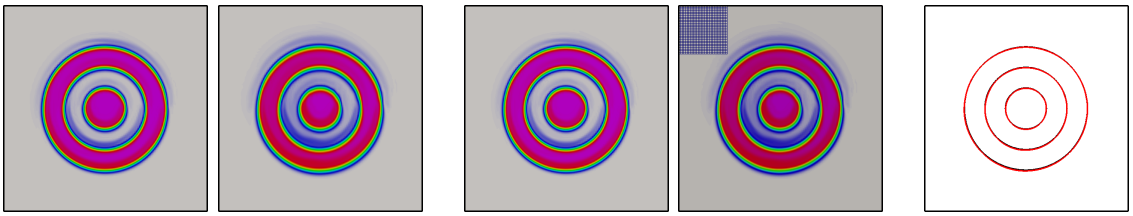
In this case we consider an initial profile and distort it via a non-periodic velocity field. The problem is given by:

$$\phi_h(\mathbf{x}, t = 0) = -\tanh\left(\frac{r - r_0}{h}\right), \tag{5.31a}$$

$$[u, v] = [-2\sin^2(\pi x)\sin(\pi y)\cos(\pi y), 2\sin^2(\pi y)\sin(\pi x)\cos(\pi x)], \tag{5.31b}$$

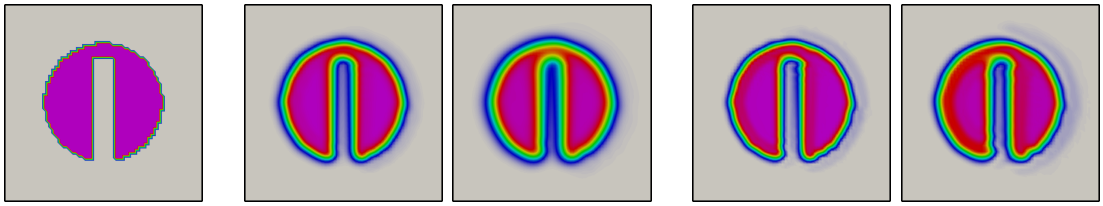
$$\Omega = [0, 1] \times [0, 1], \tag{5.31c}$$

where $r = \sqrt{(x - 0.5)^2 + (y - 0.75)^2}$ and $r_0 = 0.15$. We solve the problem using the MPP method (5.21) with $c_C = 0, 1$. In figure 5.17 we show zero contour plots (i.e., $\{\phi_h = 0\}$) at $t = 0, 1, 2, 3$ and 4. Due to the velocity profile, the initial profile gets distorted into thin regions. In these areas the numerical dissipation might produce



(a) $t = 1, 4$ with $c_E = 1$, $c_C = 1$. (b) $t = 1, 4$ with $c_E = 0.5$, $c_C = 2$. (c) $\phi_h = 0$, $t = 4$.

Figure 5.13: Refined circular rotation problem with self balanced artificial compression. We consider the MPP method (5.21) with (a) $c_E = 1$ and $c_C = 1$ and (b) $c_E = 0.5$ and $c_C = 2$ after (left) one and (right) four revolutions. In addition, we show the zero contour plots for (black) the exact solution and the solution at $t = 4$ with (green) $c_E = 1$ and $c_C = 1$ and (red) $c_E = 0.5$ and $c_C = 2$. For these simulations the mesh size is $h = 3.90 \times 10^{-3}$.



(a) Exact sol. (b) $t = 1, 4$ with $c_E = 1, c_C = 0$ (c) $t = 1, 4$ with $c_E = 1, c_C = 1$

Figure 5.14: Surface plots of the Zalesak disk problem with self balanced artificial compression. We show the (a) exact solution and the solution via the MPP method (5.21) with (b) $c_C = 0$ and (c) $c_C = 1$. For each case we show the solution at (left) $t = 1$ and (right) $t = 4$. For these simulations the mesh size is $h = 7.80 \times 10^{-3}$.

loss in the zero level set. It is clear (specially from the last panels in figure 5.17) that artificial compression helps against this problem. Finally, in figure 5.18 we repeat this numerical experiment with a refined mesh. The mesh size is given by 3.90×10^{-3}. We use in the left panel $c_E = 1$ and $c_C = 1$, in the middle panel $c_E = 0.5$ and $c_C = 2$ and in the right panel $c_E = 5$ and $c_C = 20$. We remark, once again, that increasing the compression constant produces no significant changes in the solution.

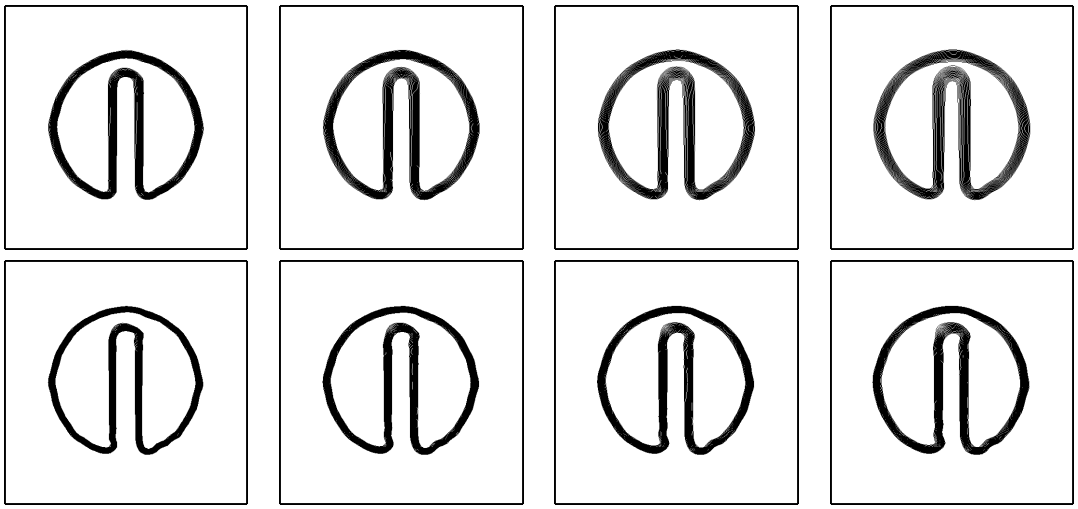
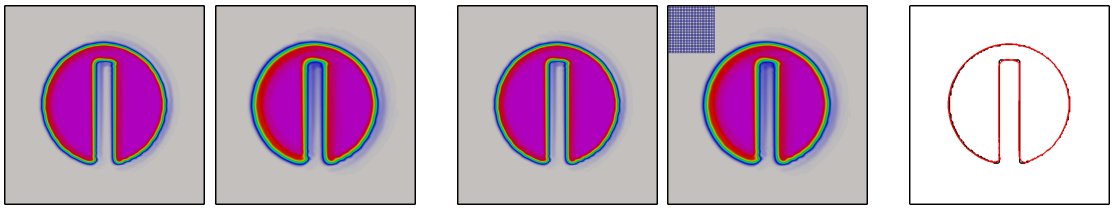


Figure 5.15: Contour plots of the Zalesak disk problem with self balanced artificial compression. We consider the MPP method (5.21) with (top) $c_E = 1$ and $c_C = 0$ and (bottom) $c_E = 1$ and $c_C = 1$ and show the solution at (from left to right) $t = 1, 2, 3, 4$. For these simulations the mesh size is $h = 7.80 \times 10^{-3}$.

5.5 Conclusions

We presented two approaches to reinitialize a smoothed Heaviside level set. The methods are mass conservative and maximum principle preserving. Both methods are one-stage methods. This means that the reinitialization process is done through the solution of the transport equation by incorporating a nonlinear anti-diffusion term based on [29, 30]. The anti-diffusion is balanced and the equation is stabilized with a nonlinear artificial viscosity based on the entropy residual of the solution following the ideas in [24, 25]. Both methods impose the maximum principle by using the flux corrected transport method by [7, 78].

The model (in the continuous level) for the first approach is given by (5.1). We emphasize that artificial dissipation and compression are added. Ideally, one desires to keep these terms balanced and, for stability reasons, to have the dissipative term to be predominant. However, if the compression constant is large enough the net effect



(a) $t = 1, 4$ with $c_E = 1$, $c_C = 1$. (b) $t = 1, 4$ with $c_E = 0.5$, $c_C = 2$. (c) $\phi_h = 0$, $t = 4$.

Figure 5.16: Refined Zalesak disk problem with self balanced artificial compression. We consider the MPP method (5.21) with (a) $c_E = 1$ and $c_C = 1$ and (b) $c_E = 0.5$ and $c_C = 2$ after (left) one and (right) four revolutions. In addition, we show the zero contour plots for (black) the exact solution and the solution at $t = 4$ with (green) $c_E = 1$ and $c_C = 1$ and (red) $c_E = 0.5$ and $c_C = 2$. For these simulations the mesh size is $h = 3.90 \times 10^{-3}$.

of these two terms can be anti-diffusion. If this happens large under/over shoots can be created and their amplitude can increase in time (see figure 5.1a). This can be avoided via flux limiting (see figure 5.1b). Having large compression might also create a non-physical behavior. This problem is depicted in (the right panel of) figure 5.9 where the zero level set is detached in thin regions. It is important to consider this effect when choosing the user parameters for specific applications. In particular, we observe a good behavior with $c_E = 1$ and $c_C = 1$ in all the experiments we performed. We use continuous Galerkin finite elements to discretize (5.1) in space. We consider first a linear viscosity as in [23]. This viscosity introduces enough dissipation to assure the solution is maximum principle preserving. The accuracy of the viscosity is later enhanced to second-order by considering a shock capturing scheme based on the entropy residual of the solution as in [24, 25]. This, however, introduces violations on the maximum principle that are later removed via the flux corrected transport. These dissipative operators are edge based instead of a (commonly used) weak formulation of the Laplace's operator. This gives the advantage of not requiring the mesh to satisfy the acute angle condition assumption. See for instance [9, 24].

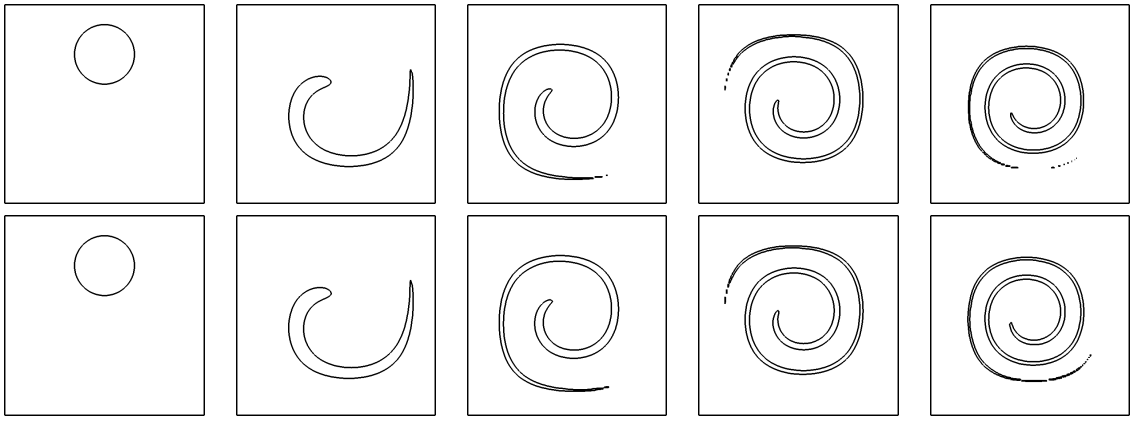
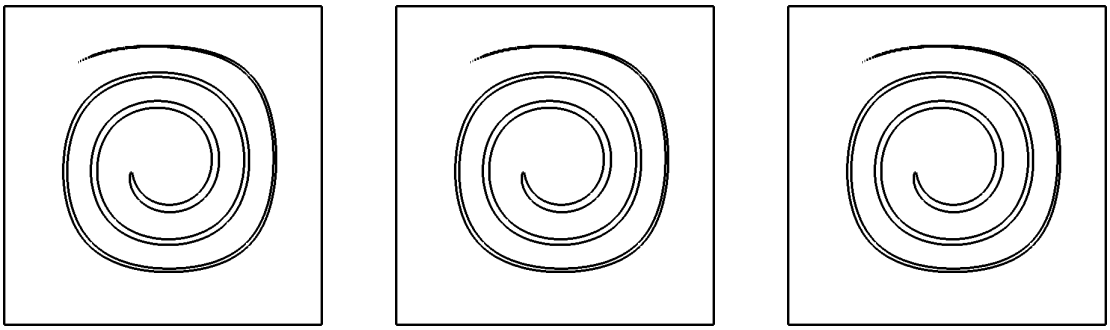


Figure 5.17: Non-periodic vortex with self balanced artificial compression. We consider the MPP method (5.21) with (top) $c_E = 1$ and $c_C = 0$ and (bottom) $c_E = 1$ and $c_C = 1$ and show the zero contour plot at (from left to right) $t = 0, 1, 2, 3, 4$. For these simulations the mesh size is $h = 7.80 \times 10^{-3}$.

The artificial compression operator is based on [29,30] and a weak formulation of the negative Laplace's operator. The main drawbacks of this method are:

- The inability of guaranteeing that the balance of artificial viscosity and compression is viscosity.
- The usage of the low-order method in [23] prevents the use of high-order spaces (see §3.2.1).
- Discretizing the artificial compression operator via a weak formulation of a negative Laplace's like operator prevents the use of arbitrary meshes.

The second approach is given by (5.15) and is designed to overcome the drawbacks of the first approach. Again artificial viscosity and compression are added to the transport equation. However the $(\cdot)^+$ operator guarantees the net effect of viscosity and compression to be viscosity. Because of this one can use large compression constants without significant change. We use continuous Galerkin finite elements to



$c_E = 1$, $c_C = 1$ $c_E = 0.5$, $c_C = 2$ $c_E = 5$, $c_C = 20$

Figure 5.18: Refined non-periodic vortex with self balanced artificial compression. We consider the MPP method (5.10) with (left) $c_E = 1$ and $c_C = 1$, (middle) $c_E = 0.5$ and $c_C = 2$ and (right) $c_E = 5$ and $c_C = 20$ and show the zero contour plot at $t = 4$. For these simulations the mesh size is $h = 3.90 \times 10^{-3}$.

discretize (5.15) in space. We consider first the linear viscosity in [26]. This operator is first-order and assures the maximum principle is preserved. In addition, we have observed the operator yields qualitatively good results with high-order spaces (see §3.2.2). The accuracy of the viscosity is improved by using a nonlinear high-order viscosity operator based on the entropy residual of the solution as proposed by [25]. Doing this, however, introduces violations on the maximum principle, which are removed by the flux corrected transport method. Finally, the artificial compression is discretized using an edge based anti-diffusive operator, which gives the advantage of not requiring the mesh to satisfy the acute angle condition.

6. NAVIER-STOKES SOLVER WITH VARIABLE DENSITY

Let $d = \{2, 3\}$ be the space dimension. We denote $\Omega \subset \mathbb{R}^d$ to be the domain with boundary $\partial\Omega \subset \mathbb{R}^d$ and consider the time interval $[0, T > 0]$. The incompressible Navier-Stokes equations with variable material parameters are given by

$$\rho(\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u}) - \nabla \cdot (\mu \nabla \mathbf{u}) + \nabla p = \mathbf{f}, \qquad \forall (\mathbf{x}, t) \in \Omega \times (0, T) \tag{6.1a}$$

$$\nabla \cdot \mathbf{u} = 0, \qquad \forall (\mathbf{x}, t) \in \Omega \times (0, T) \tag{6.1b}$$

where $\rho, \mu : \Omega \times (0, T) \to \mathbb{R}$ are the density and viscosity respectively, $\mathbf{u} : \Omega \times (0, T) \to \mathbb{R}^d$ is the velocity field, $p : \Omega \times (0, T) \to \mathbb{R}$ is the pressure and $\mathbf{f} : \Omega \times (0, T) \to \mathbb{R}^d$ is the force field.

6.1 Numerical discretization of Navier-Stokes equations

We follow a projection scheme based on [27]. The method is given as follows:

$$\rho^{n+1}\left(\frac{3\mathbf{u}^{n+1} - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\Delta t}\right) + \rho^{n+1}(\mathbf{u}^* \cdot \nabla)\mathbf{u}^{n+1} - \nabla \cdot (\mu^{n+1}\nabla \mathbf{u}^{n+1}) + \nabla p^* = \mathbf{f}^{n+1}, \tag{6.2a}$$

where $\mathbf{u}^* = 2\mathbf{u}^n - \mathbf{u}^{n-1}$ is a second-order extrapolation of $\mathbf{u}$, $p^* = p^n + \frac{4}{3}\delta\psi^n - \frac{1}{3}\delta\psi^{n-1}$ and $\delta\psi$ is a pressure correction given by:

$$-\Delta\delta\psi^{n+1} = -\frac{3\min_{\mathbf{x}}(\rho(\mathbf{x}, t = 0))}{2\Delta t}\nabla \cdot \mathbf{u}^{n+1}, \qquad \partial_n \delta\psi^{n+1}|_{\partial\Omega} = 0, \tag{6.2b}$$

$$q^{n+1} = -\mu^{n+1}\nabla \cdot \mathbf{u}^{n+1}, \tag{6.2c}$$

$$p^{n+1} = p^n + \delta\psi^{n+1} + q^{n+1}, \tag{6.2d}$$

where $\partial_n(\cdot)|_{\partial\Omega}$ is the normal derivative of $(\cdot)$ at the boundary. The initial conditions are as follows:

$$\mathbf{u}(\mathbf{x}, t=0) = \mathbf{u}_0, \tag{6.2e}$$

$$\delta\psi(\mathbf{x}, t=0), \quad q(\mathbf{x}, t=0), \quad p(\mathbf{x}, t=0) = 0. \tag{6.2f}$$

6.1.1 Spatial discretization

We use continuous Galerkin finite elements to discretize (6.2) in space. Consider a computational mesh $\mathcal{T}_h$ and define $X_h = \{\phi : \phi|_K \in \mathbb{Q}_2, \forall K \in \mathcal{T}_h, [[\phi]] = 0\}$ and $Y_h = \{\phi : \phi|_K \in \mathbb{Q}_1, \forall K \in \mathcal{T}_h, [[\phi]] = 0\}$. Let $\mathbf{u}_h, \phi \in X_h$ and $p_h, \delta\psi_h, q_h, \theta \in Y_h$. The discrete problem becomes: find $\mathbf{u}_h \in X_h$ such that

$$\int_\Omega \rho^{n+1}\left[1 + \frac{2\Delta t}{3}(\mathbf{u}_h^* \cdot \nabla)\right]\mathbf{u}_h^{n+1}\phi d\mathbf{x} + \frac{2\Delta t}{3}\int_\Omega \mu^{n+1}(\nabla\mathbf{u}_h^{n+1} \cdot \nabla\phi)d\mathbf{x} = \mathbf{b}^{n+1}(\phi), \tag{6.3a}$$

where

$$\mathbf{b}^{n+1}(\phi) := \int_\Omega \left[\rho^{n+1}\left(\frac{4}{3}\mathbf{u}_h^n - \frac{1}{3}\mathbf{u}_h^{n-1}\right) + \frac{2\Delta t}{3}\left(\mathbf{f}^{n+1} - \nabla p_h^*\right)\right]\phi d\mathbf{x}, \tag{6.3b}$$

$$\int_\Omega \nabla\delta\psi_h^{n+1} \cdot \nabla\theta d\mathbf{x} = -\frac{3\min_{\mathbf{x}}(\rho(\mathbf{x}, t=0))}{2\Delta t}\int_\Omega \nabla \cdot \mathbf{u}_h^{n+1}\theta d\mathbf{x}, \tag{6.3c}$$

$$\int_\Omega q_h^{n+1}\theta d\mathbf{x} = -\int_\Omega \mu^{n+1}\nabla \cdot \mathbf{u}_h^{n+1}\theta d\mathbf{x} \tag{6.3d}$$

$$p_h^{n+1} = p_h^n + \delta\psi_h^{n+1} + q_h^{n+1}. \tag{6.3e}$$

Note that equation (6.3a) represent a system of d uncoupled equations. Assume,

for example, $d = 3$ and let $\mathbf{u}_h = (u_h, v_h, w_h)$ and $\mathbf{b} = (b_{(x)}, b_{(y)}, b_{(z)})$. Then we obtain

$$b_{(x)}^{n+1}(\phi) := \int_\Omega [\rho^{n+1}\{1 + \frac{2\Delta t}{3}(u_h^*\partial_x + v_h^*\partial_y + w_h^*\partial_z)\}u_h^{n+1} + \frac{2\Delta t}{3}\mu^{n+1}(\partial_x\phi\partial_x + \partial_y\phi\partial_y + \partial_z\phi\partial_z)u_h^{n+1}]d\mathbf{x},$$

and similarly for v_h and w_h. Let $\{\phi_1, \ldots, \phi_N\}$ and $\{\theta_1, \ldots, \theta_M\}$, where $N = \dim(X_h)$ and $M = \dim(Y_h)$, be basis functions of X_h and Y_h respectively. Since the solutions $u_h, v_h, w_h \in X_h$ and $\delta\psi_h, q_h \in Y_h$ we can decompose them as $u_h = \sum_j U_j\phi_j$, $q_h = \sum_j Q_j\theta_j$ and so on, where U_j and Q_j are the degrees of freedom of u_h and q_h respectively. Plug these expansions into system (6.3) and let $\phi = \phi_i$ and $\theta = \theta_i$ to get:

$$AU = B_{(x)}, \tag{6.4a}$$

$$AV = B_{(y)}, \tag{6.4b}$$

$$AW = B_{(z)}, \tag{6.4c}$$

$$S\delta\Psi = F_1, \tag{6.4d}$$

$$MQ = F_2, \tag{6.4e}$$

$$P^{n+1} = P^n + \delta\Psi^{n+1} + Q^{n+1}, \tag{6.4f}$$

where we use capital letters to denote the degrees of freedom of the corresponding

variables. Here A, S and M are matrices whose ij-th elements are given by

$$\begin{aligned}
A_{ij} &= \int_\Omega [\rho^{n+1}\{1 + \frac{2\Delta t}{3}(u_h^* \partial_x + v_h^* \partial_y + w_h^* \partial_z)\}\phi_j \\
&\qquad + \frac{2\Delta t}{3}\mu^{n+1}(\partial_x \phi_j \partial_x + \partial_y \phi_j \partial_y + \partial_z \phi_j \partial_z)]\phi_i d\mathbf{x}, \\
S_{ij} &= \int_\Omega \nabla\theta_i \cdot \nabla\theta_j d\mathbf{x}, \\
M_{ij} &= \int_\Omega \theta_i \theta_j d\mathbf{x},
\end{aligned}$$

and $B_{(x)}$, F_1 and F_2 are vectors whose i-th elements are given by

$$\begin{aligned}
B_{(x),i} &= b_{(x)}(\phi_i), \\
F_{1,i} &= -\frac{3\min_{\mathbf{x}}(\rho(\mathbf{x}, t=0))}{2\Delta t}\int_\Omega \nabla \cdot \mathbf{u}_h^{n+1}\theta_i d\mathbf{x}, \\
F_{2,i} &= -\int_\Omega \mu^{n+1}\nabla \cdot \mathbf{u}_h^{n+1}\theta_i d\mathbf{x},
\end{aligned}$$

and similarly for $B_{(y)}$ and $B_{(z)}$.

6.1.1.1 Validation of Navier-Stokes solver

In this section we perform convergence studies for method (6.3). We define a 2D computational domain to be $\Omega = (0,1) \times (0,1)$ and an exact solution given by

$$\mathbf{u}(\mathbf{x}, t) = (\sin(x)\sin(y+t), \cos(x)\cos(y+t)), \tag{6.5a}$$

$$p(\mathbf{x}, t) = \cos(x)\sin(y+t), \tag{6.5b}$$

$$\rho(\mathbf{x}, t) = \sin^2(x+y+t) + 1, \tag{6.5c}$$

$$\mu(\mathbf{x}, t) = 1, \tag{6.5d}$$

and compute the corresponding force term via (6.1). Note that $\nabla \cdot \mathbf{u} = 0, \forall(\mathbf{x}, t) \in (\Omega \times (0, T))$. The force term $\mathbf{f} = (f^{(x)}, f^{(y)})$ is given by:

$$f^{(x)} = x\sin(x)\{\sin(t+y) + \cos(x)[1+\sin^2(t+x+y)] + \cos(t+y)[1+\sin^2(t+x+y)]\} \tag{6.6a}$$

$$f^{(y)} = 3\cos(x)\cos(t+y) - 0.5\sin[2(t+y)][1+\sin(t+x+y)] - \cos(x)\sin(t+y)[1+\sin^2(t+x+y)] \tag{6.6b}$$

We perform convergence studies in space and time and obtain the results in tables 6.1 and 6.2 respectively. We observe similar convergence rates as reported in the mentioned reference; i.e., we observe close to second-order convergence in time.

h	$\|\|E(\mathbf{u})\|\|_{L^2}$	Rate	$\|\|E(\mathbf{u})\|\|_{H^1}$	Rate
6.25E-2	1.04E-4		8.94E-4	
3.13E-2	1.51E-5	2.78	1.94E-4	2.20
1.56E-2	2.05E-6	2.88	4.50E-5	2.11
7.81E-3	2.67E-7	2.93	1.09E-5	2.04
h	$\|\|E(p)\|\|_{L^2}$	Rate	$\|\|E(p)\|\|_{H^1}$	Rate
6.25E-2	9.71E-4		4.22E-2	
3.13E-2	1.99E-4	2.28	2.09E-2	1.01
1.56E-2	4.47E-5	2.15	1.04E-2	1.00
7.81E-3	1.07E-5	2.06	5.21E-03	0.99

Table 6.1: Convergence in space of Navier-Stokes solver (6.2). Here $E(\cdot)$ denotes the error of $(\cdot)$.

6.2 Air flow through low-pressure turbine blades

We test the previously revisited Navier-Stokes solver to simulate one-phase fluid flow at high Reynolds numbers through turbine blades. This work is done in collabora-

Δt	$\|\|E(\mathbf{u})\|\|_{L^2}$	Rate	$\|\|E(\mathbf{u})\|\|_{H^1}$	Rate
1.00E-1	3.12E-3		1.56E-2	
5.00E-2	1.02E-3	1.61	5.26E-3	1.56
2.50E-2	2.93E-4	1.80	1.55E-3	1.75
1.25E-2	7.95E-5	1.88	4.43E-4	1.81
6.25E-3	2.09E-5	1.92	1.24E-4	1.83
Δt	$\|\|E(p)\|\|_{L^2}$	Rate	$\|\|E(p)\|\|_{H^1}$	Rate
1.00E-1	2.01E-2		1.21E-1	
5.00E-2	6.61E-3	1.60	5.81E-2	1.05
2.50E-2	1.93E-3	1.77	2.36E-2	1.29
1.25E-2	5.36E-4	1.84	9.26E-3	1.34
6.25E-3	1.45E-4	1.88	3.79E-3	1.28

Table 6.2: Convergence in time of Navier-Stokes solver(6.2). Here $E(\cdot)$ denotes the error of $(\cdot)$.

tion with Prof. Meinhard T. Schobeiri and his Ph.D. student Ali Nikparto from the Mechanical Engineering department at Texas A&M University. The objective is to obtain Direct Numerical Simulations (DNS) of a single fluid moving around a set of turbine blades accommodated in cascade, see figure 6.1. The material properties are

$$\rho = 1.185, \tag{6.7}$$

$$\mu = 1.831 \times 10^{-5}, \tag{6.8}$$

with a velocity magnitude at the inlet (left) boundary of $U = 3.9284$. This corresponds to a Reynolds number of $Re \approx 1.6 \times 10^5$, which is computed by considering $L = 0.25$ and $U_{\max} = 10$. In figure 6.2 we show different sections of the computational grid.

Due to computational restrictions we couldn't afford running simulations with much finer grids. The finest grid we have has a smallest mesh size of $h = 3.01 \times 10^{-4}$. When the Reynolds number is large, physical small scales in the velocity are present. In this situation, not having a fine enough mesh introduces instabilities. In particular,

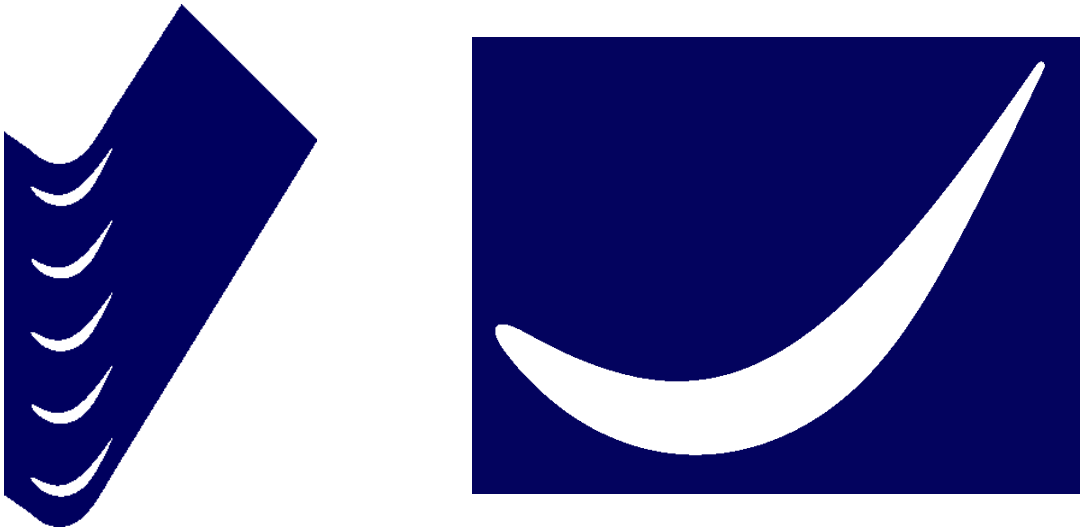


Figure 6.1: Low pressure turbine blades. We show (left) multiple blades in cascade and (right) a zoomed single blade.

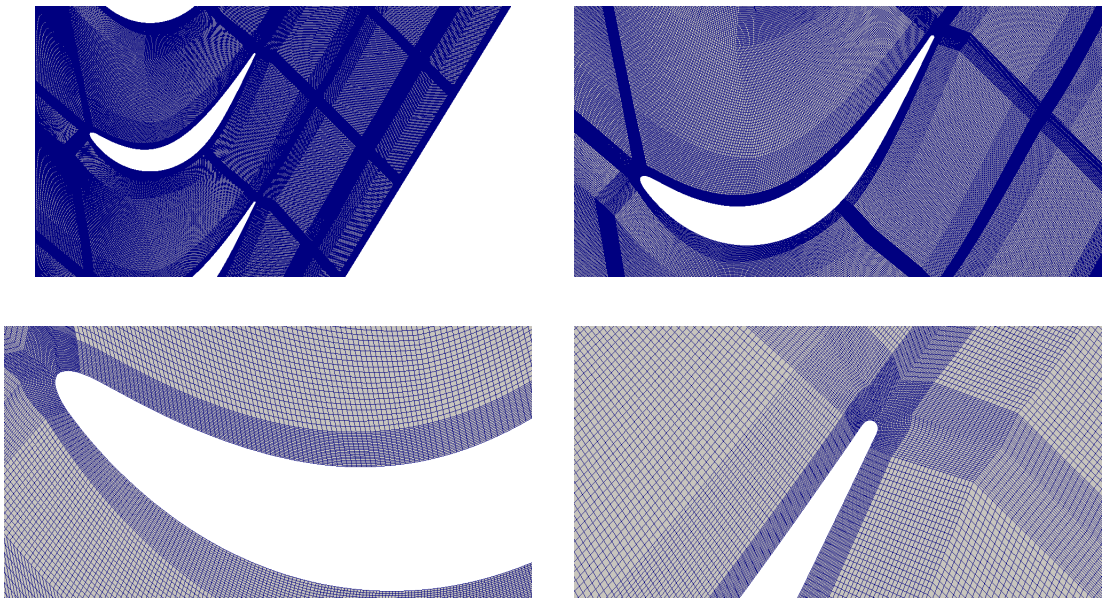


Figure 6.2: Zoomed low pressure turbine blades. We show multiple zoomed-in pictures of the computational grid.

if the cell-Reynolds number $\frac{\rho U h}{\mu}$ is larger than a small multiple of one, instabilities appear in the solution, see for instance [22]. We observe these instabilities with the parameters we aim for and the finest mesh we have. There is an extensive list of methodologies to reduce these instabilities. See for instance [3, 8, 40, 44] for the so called Galerkin Least Squares (GaLS) stabilization methods and variations to it, [45, 62] for the use of grad div stabilization, [21,22] for stabilization via subgrid modeling, [39, 43] for multiscale methods; in addition, discontinuous Galerkin methods [55, 67] can be used. Nevertheless, all these methods are introducing artificial dissipation in some form; i.e., they are not Direct Numerical Simulations (DNS). To solve the problem without extra stabilization terms (i.e., to perform DNS) we require to refine the computational grid, which is something out of our computational capabilities. The largest Reynolds number we could afford to run DNS is of $Re \approx 5.393 \times 10^4$ (one third of the objective). In figure 6.3 we show the magnitude of the velocity field at $T = 1.5$ for such Reynolds number.

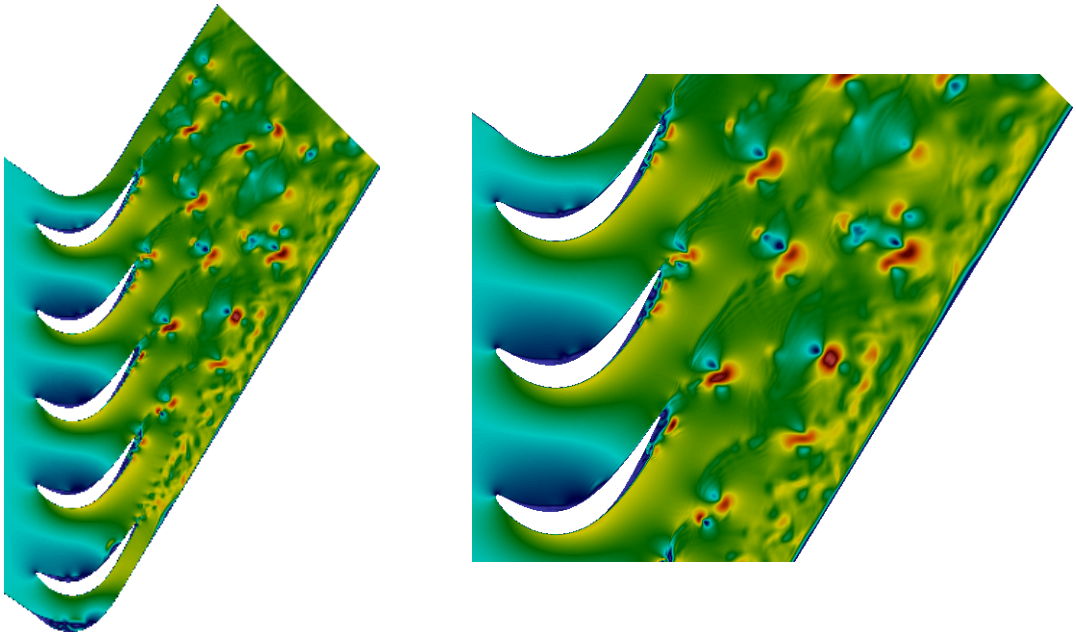


Figure 6.3: Navier-Stokes velocity on low pressure turbine blades. We show the velocity magnitude at $T = 1.5$ for $Re \approx 5.393 \times 10^4$.

7. MULTIPHASE FLOW

In this section we use the maximum principle preserving method with artificial compression presented in §5.3.6.1 along with an incompressible Navier-Stokes solver to simulate two-phase flow. The method used to solve the Navier-Stokes equations is documented in [27] and revisited in chapter 6.

7.1 Overview of the methodology

The methodology for solving the two-phase field problem is as follows. For an initial state of density and viscosity fields we compute the velocity and pressure by solving the Navier-Stokes equations, following [27]. This velocity field is used to transport the level set, using the method in §5.3.6.1. Afterwards, the level set function is used to reconstruct density and viscosity fields as follows:

$$\rho = \rho_{\text{water}} \left(\frac{1 + H_\epsilon(\phi)}{2} \right) + \rho_{\text{air}} \left(\frac{1 - H_\epsilon(\phi)}{2} \right), \tag{7.1a}$$

$$\mu = \mu_{\text{water}} \left(\frac{1 + H_\epsilon(\phi)}{2} \right) + \mu_{\text{air}} \left(\frac{1 - H_\epsilon(\phi)}{2} \right), \tag{7.1b}$$

where $\phi \in [-1, 1]$ is the level set function and H_ϵ is a regularized Heaviside function. In particular, the set $\{\phi = 1\}$ represents water and the set $\{\phi = -1\}$ represents air. Ideally, the transition from -1 to 1 should be maintained as sharp as possible since values of ϕ in the range $(-1, 1)$ give non-physical values for the reconstructed fields.

The regularized Heaviside is given by

$$H(\phi) = \begin{cases} 1 & \text{if } \phi > \epsilon \\ -1 & \text{if } \phi < -\epsilon \\ \phi/\epsilon & \text{otherwise} \end{cases}. \tag{7.2}$$

All the experiments reported in this section are done with quadrangular meshes and $\epsilon = h$ where $h = \Delta x = \Delta y$ is the cell size. The velocity is approximated using $\mathbb{Q}_2$ elements and the pressure is approximated using $\mathbb{Q}_1$ elements. This process is repeated until the final time is reached.

For all experiments in this chapter the material parameters are given by

$$\rho_{\text{water}} = 1000, \quad \rho_{\text{air}} = 1, \quad \mu_{\text{water}} = 1, \quad \mu_{\text{air}} = 1 \times 10^{-2}, \tag{7.3}$$

and the gravity coefficient is $g = -1$.

7.2 Two-dimensional falling drop

In this problem we consider a drop of water, surrounded by air, falling towards water in rest. The domain is given by $\Omega = [0, 0.3] \times [0, 0.9]$. The initial condition consists of water and air occupying the domain

$$W = \{(x, y) \mid y \leq 0.2\} \cup \{(x, y) \mid \sqrt{(x - 0.15)^2 + (y - 0.75)^2} \leq 0.1\}$$
$$A = \Omega \setminus W.$$

Both water and air are at rest at the initial time. At $t = 0$ the system evolve under the action of gravity; i.e., the drop of water falls hitting the water in rest in the bottom of the domain. We impose $\mathbf{u} = \mathbf{0}$ at all the boundaries; i.e., we consider the no-slip

boundary condition. The mesh size of the spatial discretization is $h = 2.343 \times 10^{-3}$. In figure 7.1 we show the zero level set $\{\phi = 0\}$ for different times.

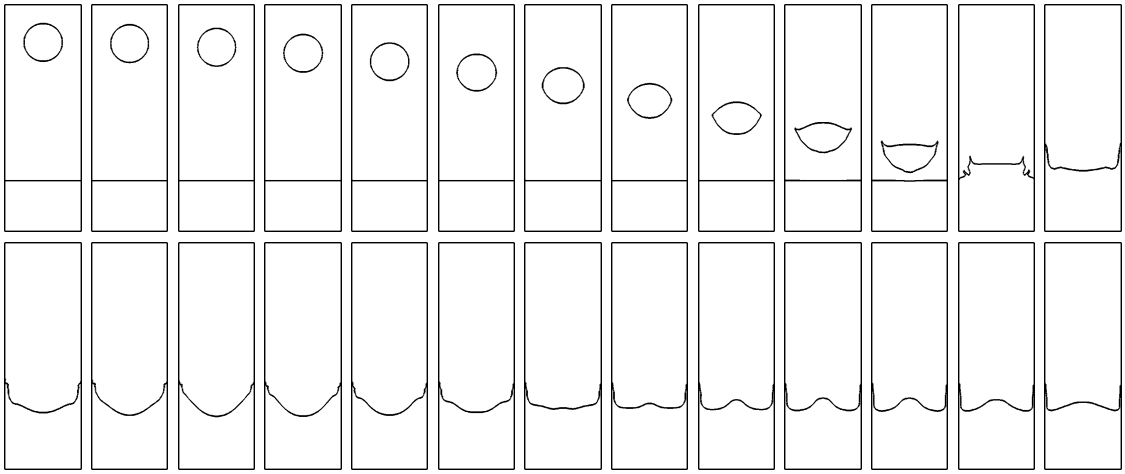


Figure 7.1: Two dimensional falling drop problem. We consider the MPP method (5.21) with $c_E = 1$, $c_C = 1$. The times shown are (from left to right and top to bottom) $t = [0, 0.2, \ldots 5]$.

7.3 Two-dimensional dam breaking

We consider the two-dimensional dam breaking problem on a domain defined by $\Omega = [0, 1] \times [0, 0.5]$. The initial data consists of water and air occupying the domain

$$W = \{(x, y) \mid |x - 0.5| \leq 0.15, \;\; y \leq 0.35\}$$
$$A = \Omega \backslash W.$$

Both water and air are at rest at the initial time. At $t = 0$ we let the system evolve under the action of gravity; i.e., the column of water collapses under its own weight and spreads over the tank. We impose $\mathbf{u} = 0$ at all the boundaries; i.e., we consider

the no-slip boundary condition. The mesh size of the spatial discretization used is $h = 1.953 \times 10^{-3}$. The zero level set $\{\phi = 0\}$ is depicted, for various times, in figure 7.2.

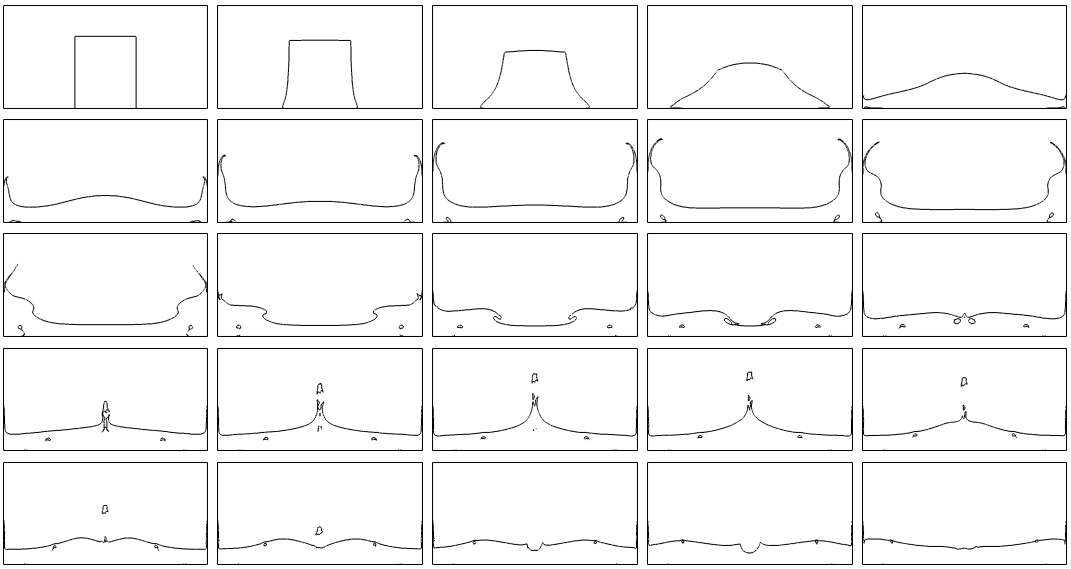


Figure 7.2: Two dimensional dam breaking problem. We consider the MPP method (5.21) with $c_E = 1$, $c_C = 1$. The times shown are (from left to right and top to bottom) $t = [0, 0.2, \ldots, 4.8]$.

7.4 Two-dimensional tank filling

The tank-filling test problem simulates water entering a tank filled with air. The domain is given by $\Omega = [0, 0.4] \times [0, 0.4]$. The initial data consists of water and air occupying the domain

$$W = \{(x, y) \mid x \leq 0.01,\ |y - 0.325| \leq 0.025\},$$
$$A = \Omega \setminus W.$$

Both water and air are at rest at the initial time. The boundary conditions on the velocity are:

$$\mathbf{u} = \mathbf{u}_{\text{left}} := \begin{cases} (0.25, 0), & \forall x = 0,\ y \in [0.3, 0.35], \\ (0, 0), & \forall x = 0,\ y \notin [0.3, 0.35], \end{cases}$$

$$\mathbf{u} = \mathbf{u}_{\text{top}} := \begin{cases} (0, 0.25), & \forall x \in [0.3, 0.35],\ y = 0.4, \\ (0, 0), & \forall x \notin [0.3, 0.35],\ y = 0.4, \end{cases}$$

$\mathbf{u} =: \mathbf{u}_{\text{right}} = 0$ if $x = 0.4$ and $\mathbf{u} =: \mathbf{u}_{\text{bottom}} = 0$ if $y = 0$. The boundary condition for the level set is: $\phi = 1, \forall x = 0,\ y \in [0.3, 0.35]$; i.e., water is introduced. The mesh size of the spatial discretization used is $h = 1.562 \times 10^{-3}$. The zero level set $\{\phi = 0\}$ is shown, for various times, in figure 7.3.

7.5 Three-dimensional tank filling

Now we simulate a three dimensional version of the tank-filling problem. The domain is given by $\Omega = [0, 0.4] \times [0, 0.4] \times [0, 0.1]$. The initial data consists of water occupying the domain $W = \{(x, y, z) \mid x \leq 0.01,\ |y - 0.325| \leq 0.025, |z - 0.05| \leq 0.025\}$ and air occupying the domain $A = \Omega \setminus W$. Both water and air are at rest at the initial time. The boundary conditions on the velocity are:

$$\mathbf{u} = \mathbf{u}_{\text{left}} := \begin{cases} (0.25, 0, 0), & \forall x = 0,\ y \in [0.3, 0.35], z \in [0.025, 0.075], \\ (0, 0, 0), & \forall x = 0,\ y \notin [0.3, 0.35], z \notin [0.025, 0.075], \end{cases}$$

$$\mathbf{u} = \mathbf{u}_{\text{top}} := \begin{cases} (0, 0.25, 0), & \forall x \in [0.3, 0.35],\ y = 0.4, z \in [0.025, 0.075], \\ (0, 0, 0), & \forall x \notin [0.3, 0.35],\ y = 0.4, z \notin [0.025, 0.075], \end{cases}$$

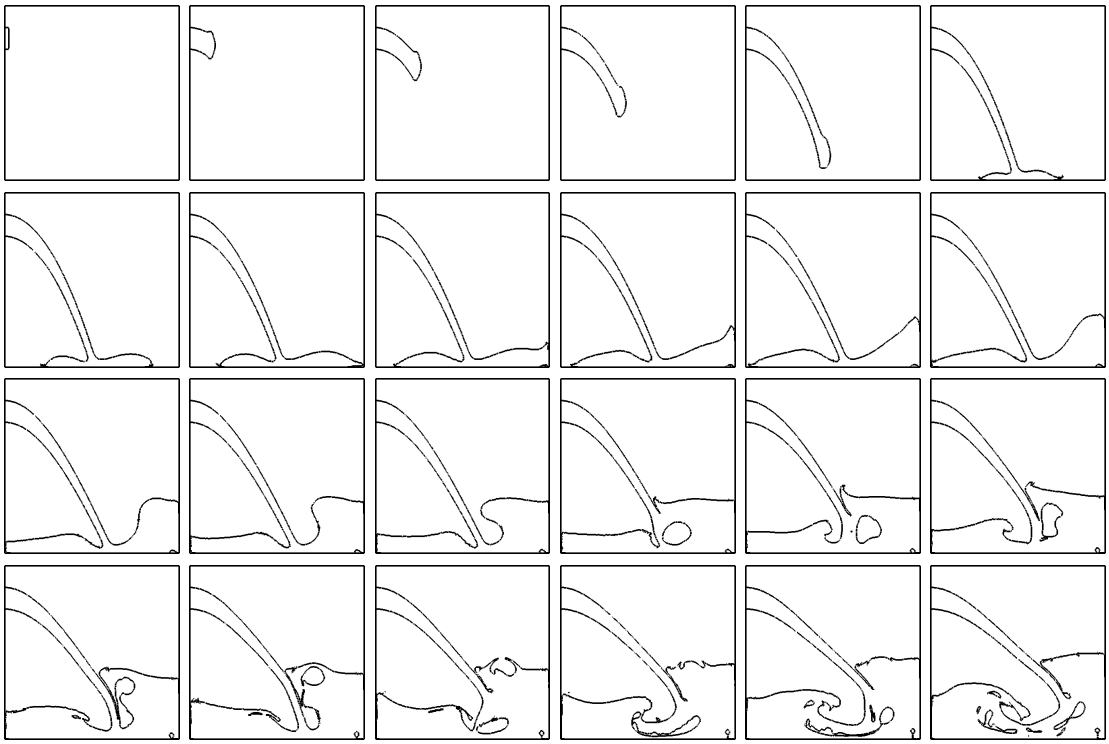


Figure 7.3: Two dimensional filling tank. We consider the MPP method (5.21) with $c_E = 1$, $c_C = 1$. The times shown are (from left to right and top to bottom) $t = [0, 0.2, \ldots, 4.6]$.

$\mathbf{u} =: \mathbf{u}_{\text{right}} = 0$ if $x = 0.4$, $\mathbf{u} =: \mathbf{u}_{\text{bottom}} = 0$ if $y = 0$, $\mathbf{u} =: \mathbf{u}_{\text{front}} = 0$ if $z = 0$ and $\mathbf{u} =: \mathbf{u}_{\text{back}} = 0$ if $z = 0.1$. The boundary condition for the level set is: $\phi = 1, \forall x = 0,\ y \in [0.3, 0.35],\ z \in [0.025, 0.075]$; i.e., water is introduced. The mesh size of the spatial discretization used is $h = 7.654 \times 10^{-3}$. The zero level set $\{\phi = 0\}$ is shown, for various times, in figure 7.4.

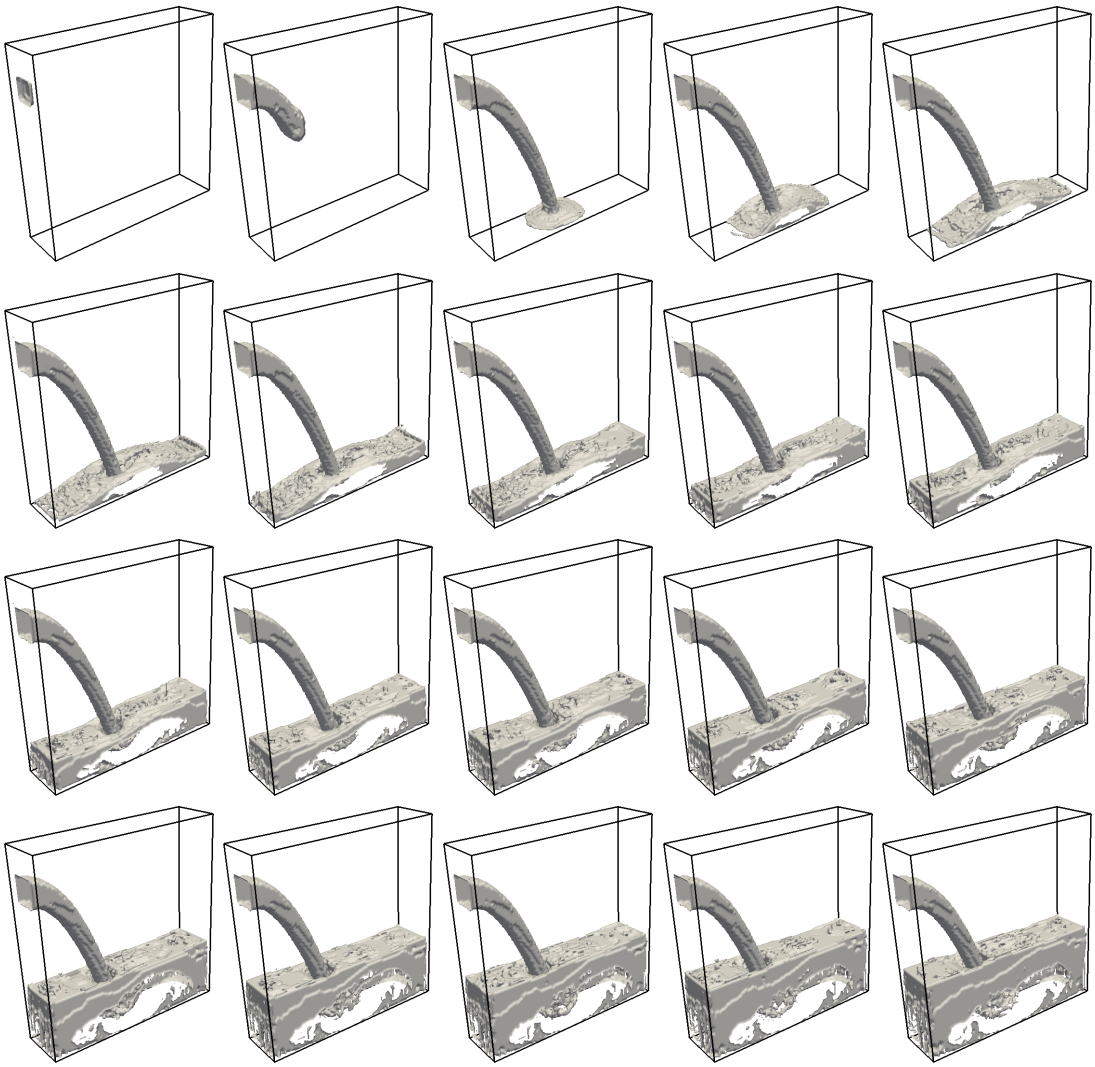


Figure 7.4: Three dimensional filling tank. We consider the MPP method (5.21) with $c_E = 1$, $c_C = 1$. The times shown are (from left to right and top to bottom) $t = [0, 0.5, \ldots, 9.5]$.

8. CONCLUSIONS

In this work we present methods to solve the linear conservation law preserving the maximum principle. To do this we consider continuous and discontinuous Galerkin finite elements. We use high-order polynomial spaces with positive basis functions given by Bernstein polynomials.

In chapter 3 we used the standard Flux Corrected Transport (FCT) method with continuous Galerkin finite elements. By doing this we obtained the expected (and optimal) high-order accuracy with $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_3$ spaces when we used smooth solutions that are monotone. However, non-physical oscillations are introduced with high-order spaces. This is true nevertheless the method is maximum principle preserving. To eliminate this behavior we considered a localized stencil to compute the bounds. This process reduced the oscillatory behavior but introduced dissipation for the higher-order spaces. However, the method preserves its convergence properties.

In chapter 4 we considered discontinuous Galerkin finite element spaces and proposed two methods. The first approach consists on applying the FCT method with localized bounds. By doing this we reduced oscillatory behavior when high-order spaces are used. Similar than with continuous spaces, using localized bounds leads to more dissipative solutions. However, this method recovers the expected (and optimal) high-order accuracy with $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_3$ spaces for smooth solutions that are monotone. The second approach presents a shift in the paradigm used on the "classical" FCT methodology. Instead of using a mass conservative low-order Maximum Principle Preserving (MPP) and a mass conservative high-order non-MPP solution we use the same low-order solution and a non mass conservative MPP solution. Then, we interpolate from the low- to the high-order solution using a single

parameter designed to recover mass conservation. By solving a nonlinear problem per cell we localize this process inside a cell to the level of degrees of freedom. We obtained clearly less dissipated solutions using this approach with high-order spaces. The expected high-order accuracy is also recovered with $\mathbb{Q}_1$, $\mathbb{Q}_2$ and $\mathbb{Q}_3$ for smooth solutions that are monotone.

The optimal accuracy with all methods in chapters 3 and 4 is lost around local extrema. No better than second-order (in the L^1 norm) was obtained regardless of the polynomial space. This is a common problem with methods that impose some type of monotonicity constraint. There are alternatives (mainly within finite volume methods) to solve this issue. Most of these methods relax the monotonicity constraint around local extrema. This is a possible direction to improve the methods presented in this work.

In chapter 5, we used one of the methods presented in chapter 3 to transport a smoothed Heaviside level set function with a one-stage reinitialization based on artificial sharpening. We proposed two alternatives. The first method allows for large compressions that yield sharp solutions but might introduce non-physical behavior (if excessive compression is used). To overcome this problem we proposed a second approach that controls the strength of the artificial compression.

In chapter 6 we revisited a projection method to solve the incompressible Navier-Stokes equations by [27] for variable material density. We validated the implementation via convergence tests in space and time. In addition, we used this Navier-Stokes solver to simulate one-phase flow around low pressure turbine blades at (relatively) large Reynolds numbers.

Finally, in chapter 7 we used one of the methods in chapter 5 to transport a level set function and the Navier-Stokes solver revisited in chapter 6 to solve two-phase incompressible flows in two and three dimensions.

REFERENCES

[1] J. Ahrens, B. Geveci, C. Law, C. Hansen, and C. Johnson. ParaView: An End-User Tool for Large-Data Visualization. Visualization Handbook, pages 717–731, 2005.

[2] R. Anderson, V. Dobrev, T. V. Kolev, and R. Rieben. Monotonicity in high-order curvilinear finite element Arbitrary Lagrangian-Eulerian remap. International Journal for Numerical Methods in Fluids, 77(5):249–273, 2015.

[3] C. Baiocchi, F. Brezzi, and L. P. Franca. Virtual bubbles and Galerkin-Least-Squares type methods (Ga. LS). Computer Methods in Applied Mechanics and Engineering, 105(1):125–141, 1993.

[4] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II – a General Purpose Object Oriented Finite Element Library. ACM Transactions on Mathematical Software, 33(4):24/1–24/27, 2007.

[5] D. J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. Computer Methods in Applied Mechanics and Engineering, 99(2):235–394, 1992.

[6] A. Bonito, J.-L. Guermond, and S. Lee. Numerical simulations of bouncing jets. International Journal for Numerical Methods in Fluids, 80(1):53–75, 2016.

[7] J. P. Boris and D. L. Book. Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works. Journal of Computational Physics, 11(1):38–69, 1973.

[8] A. N. Brooks and T. J. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incom-

pressible Navier-Stokes equations. Computer Methods in Applied Mechanics and Engineering, 32(1):199–259, 1982.

[9] E. Burman and A. Ern. Nonlinear diffusion and discrete maximum principle for stabilized Galerkin approximations of the convection-diffusion-reaction equation. Computer Methods in Applied Mechanics and Engineering, 191(35):3833–3855, 2002.

[10] P.-H. Chiu and Y.-T. Lin. A conservative phase field method for solving incompressible two-phase flows. Journal of Computational Physics, 230:185–204, 2011.

[11] M. G. Crandall and A. Majda. Monotone difference approximations for scalar conservation laws. Mathematics of Computation, 34(149):1–21, 1980.

[12] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodriguez-Ferran. Arbitrary Lagrangian-Eulerian methods. Encyclopedia of Computational Mechanics Vol. 1: Fundamentals., Chapter 14.

[13] L. J. Durlofsky, B. Engquist, and S. Osher. Triangle based adaptive stencils for the solution of hyperbolic conservation laws. Journal of Computational Physics, 98(1):64–73, 1992.

[14] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. Journal of Computational Physics, 183(1):83–116, 2002.

[15] R. Fedkiw, G. Sapiro, and C.-W. Shu. Shock capturing, level sets, and PDE based methods in computer vision and image processing: a review of Osher's contributions. Journal of Computational Physics, 185:309–341, 2003.

[16] R. Garimella, M. Kuchařík, and M. Shashkov. Efficient algorithm for local-bound-preserving remapping in ALE methods. In Numerical Mathematics and Advanced Applications, pages 358–367. Springer, 2004.

[17] F. Gibou, R. Fedkiw, R. Caflisch, and S. Osher. A level set approach for the numerical simulation of dendritic growth. Journal of Scientific Computing, 19:183–199, 2003.

[18] GLVis. Accurate finite element visualization. `www.glvis.org`. Visited on 06/02/2015.

[19] S. K. Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. Matematicheskii Sbornik, 89(3):271–306, 1959.

[20] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. SIAM Review, 43(1):89–112, 2001.

[21] J.-L. Guermond. Stabilization of Galerkin approximations of transport equations by subgrid modeling. ESAIM: Mathematical Modelling and Numerical Analysis, 33(06):1293–1316, 1999.

[22] J.-L. Guermond, A. Marra, and L. Quartapelle. Subgrid stabilized projection method for 2D unsteady flows at high Reynolds numbers. Computer Methods in Applied Mechanics and Engineering, 195(44):5857–5876, 2006.

[23] J.-L. Guermond and M. Nazarov. A maximum-principle preserving C0 finite element method for scalar conservation equations. Computer Methods in Applied Mechanics and Engineering, 272:198–213, 2014.

[24] J.-L. Guermond, M. Nazarov, B. Popov, and Y. Yang. A second-order maximum principle preserving lagrange finite element technique for nonlinear scalar

conservation equations. SIAM Journal on Numerical Analysis, 52(4):2163–2182, 2014.

[25] J.-L. Guermond, R. Pasquetti, and B. Popov. Entropy viscosity method for nonlinear conservation laws. Journal of Computational Physics, 230(11):4248–4267, 2011.

[26] J.-L. Guermond and B. Popov. Invariant domains and first-order continuous finite element approximation for hyperbolic systems. arXiv, 2015. arXiv:1509.07461.

[27] J.-L. Guermond and A. Salgado. A splitting method for incompressible flows with variable density based on a pressure Poisson equation. Journal of Computational Physics, 228(8):2834–2846, 2009.

[28] E. Hachem, S. Feghali, and T. Coupez. Anisotropic adaptive meshing and monolithic variational multiscale method for fluid-struction interaction. Computers and Structures, 122:88–100, 2013.

[29] A. Harten. The artificial compression method for computation of shocks and contact discontinuities. I. Single conservation laws. Communications on Pure and Applied Mathematics, 30(5):611–638, 1977.

[30] A. Harten. The artificial compression method for computation of shocks and contact discontinuities. III. Self-adjusting hybrid schemes. Mathematics of Computation, 32:363–389, 1978.

[31] A. Harten. High resolution schemes for hyperbolic conservation laws. Journal of Computational Physics, 49(3):357–393, 1983.

[32] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. Journal of Computational

Physics, 71(2):231–303, 1987.

[33] A. Harten, J. M. Hyman, P. D. Lax, and B. Keyfitz. On finite-difference approximations and entropy conditions for shocks. Communications on Pure and Applied Mathematics, 29(3):297–322, 1976.

[34] A. Harten and S. Osher. Uniformly high-order accurate nonoscillatory schemes. I. SIAM Journal on Numerical Analysis, 24(2):279–309, 1987.

[35] A. Harten, S. Osher, B. Engquist, and S. R. Chakravarthy. Some results on uniformly high-order accurate essentially nonoscillatory schemes. Applied Numerical Mathematics, 2(3):347–377, 1986.

[36] A. Harten and G. Zwas. Self-adjusting hybrid schemes for shock computations. Journal of Computational Physics, 9(3):568–583, 1972.

[37] C. Hirt, A. A. Amsden, and J. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. Journal of Computational Physics, 14(3):227–253, 1974.

[38] C. W. Hirt and B. D. Nichols. Volume of Fluid (VOF) method for the dynamics of free boundaries. Journal of Computational Physics, 39(1):201–225, 1981.

[39] T. J. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. Computer Methods in Applied Mechanics and Engineering, 127(1):387–401, 1995.

[40] T. J. Hughes, L. P. Franca, and G. M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. Computer Methods in Applied Mechanics and Engineering, 73(2):173–189, 1989.

[41] T. J. Hughes, W. K. Liu, and T. K. Zimmermann. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. Computer Methods in Applied Mechanics and Engineering, 29(3):329–349, 1981.

[42] S. Ianniello and A. Di Mascio. A self-adaptive oriented particles Level-Set method for tracking interfaces. Journal of Computational Physics, 229(4):1353–1380.

[43] V. John, S. Kaya, and W. Layton. A two-level variational multiscale method for convection-dominated convection–diffusion equations. Computer Methods in Applied Mechanics and Engineering, 195(33):4594–4603, 2006.

[44] C. Johnson, U. Nävert, and J. Pitkäranta. Finite element methods for linear hyperbolic problems. Computer Methods in Applied Mechanics and Engineering, 45(1):285–312, 1984.

[45] G. Kobelakov. On solving the Navier-Stokes equations at large Reynolds numbers. Russian Journal of Numerical Analysis and Mathematical Modelling, 10(1):33–40, 1995.

[46] D. Kuzmin. A high-resolution finite element scheme for convection-dominated transport. Communications in Numerical Methods in Engineering, 16(3):215–223, 2000.

[47] D. Kuzmin, R. Löhner, and S. Turek. Flux-Corrected Transport: Principles, Algorithms, and Applications. Scientific Computation. Springer, 2005.

[48] D. Kuzmin and M. Möller. Algebraic Flux Correction I. Scalar Conservation Laws. Springer, 2005.

[49] D. Kuzmin, M. Möller, and S. Turek. High-resolution FEM–FCT schemes for multidimensional conservation laws. Computer Methods in Applied Mechanics

and Engineering, 193(45):4915–4946, 2004.

[50] D. Kuzmin and F. Schieweck. A parameter-free smoothness indicator for high-resolution finite element schemes. Central European Journal of Mathematics, 11(8):1478–1488, 2013.

[51] D. Kuzmin and S. Turek. Flux correction tools for finite elements. Journal of Computational Physics, 175(2):525–558, 2002.

[52] D. Kuzmin and S. Turek. High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter. Journal of Computational Physics, 198(1):131–158, 2004.

[53] P. Lax and B. Wendroff. Systems of conservation laws. Communications on Pure and Applied Mathematics, 13:217–237, 1960.

[54] P. D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. Communications on Pure and Applied Mathematics, 7(1):159–193, 1954.

[55] P. Lesaint and P.-A. Raviart. On a finite element method for solving the neutron transport equation. Mathematical Aspects of Finite Elements in Partial Differential Equations, (33):89–123, 1974.

[56] R. J. LeVeque. Numerical Methods for Conservation Laws, volume 132. Springer, 1992.

[57] X.-D. Liu. A maximum principle satisfying modification of triangle based adapative stencils for the solution of scalar hyperbolic conservation laws. SIAM Journal on Numerical Analysis, 30(3):701–716, 1993.

[58] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. Journal of Computational Physics, 115(1):200–212, 1994.

[59] L. E. Malvern. Introduction to the Mechanics of a Continuous Medium. Number Monograph. 1969.

[60] MFEM. Modular parallel finite element methods library. `http://mfem.org/`. Visited on 06/02/2015.

[61] W. Mulder and S. Osher. Computing interface motion in compressible Gas Dynamics. Journal of Computational Physics, 100:209–228, 1990.

[62] M. Olshanskii and A. Reusken. Grad-div stablilization for Stokes equations. Mathematics of Computation, 73(248):1699–1718, 2004.

[63] E. Olsson and G. Kreiss. A conservative level set method for two phase flow. Journal of Computational Physics, 210:225–246, 2005.

[64] S. Osher. Convergence of generalized MUSCL schemes. SIAM Journal on Numerical Analysis, 22(5):947–961, 1985.

[65] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics, 79(1):12–49, 1988.

[66] G. M. Phillips. Interpolation and Approximation by Polynomials. Springer Science & Business Media.

[67] W. H. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Los Alamos Report LA-UR-73-479, 1973.

[68] C. Schär and P. K. Smolarkiewicz. A synchronous and iterative flux-correction formalism for coupled transport equations. Journal of Computational Physics, 128(1):101–120, 1996.

[69] M. Sussman, A. Almgren, J. Bell, P. Colella, L. Howell, and M. Welcome. An Adaptive Level Set Approach for Incompressible Two-Phase Flows. Journal of Computational Physics, 148:81–124, 1999.

[70] M. Sussman and E. Puckett. A coupled level set and Volume-Of-Fluid method for computing 3D and axisymmetric incompressible two-phase flows. Journal of Computational Physics, 162:301–337, 2000.

[71] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. Journal of Computational Physics, 114(1):146–159, 1994.

[72] B. van Leer. Towards the ultimate conservative difference scheme I. The quest of monotonicity. Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics, pages 163–168, 1973.

[73] B. Van Leer. Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme. Journal of Computational Physics, 14(4):361–370, 1974.

[74] B. Van Leer. Towards the ultimate conservative difference scheme III. Upstream-centered finite-difference schemes for ideal compressible flow. Journal of Computational Physics, 23(3):263–275, 1977.

[75] B. Van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. Journal of Computational Physics, 32(1):101–136, 1979.

[76] L. Ville, L. Silva, and T. Coupez. Convected level set method for the numerical simulation of fluid buckling. International Journal for Numerical Methods in Fluids, 66(3):324–344, 2011.

[77] J. Von Neumann and R. D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. Journal of Applied Physics, 21(3):232–237, 1950.

[78] S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. Journal of Computational Physics, 31(3):335–362, 1979.

[79] X. Zhang and C.-W. Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. Journal of Computational Physics, 229(9):3091–3120, 2010.

[80] X. Zhang, Y. Xia, and C.-W. Shu. Maximum-principle-satisfying and positivity-preserving high order discontinuous Galerkin schemes for conservation laws on triangular meshes. Journal of Scientific Computing, 50(1):29–62, 2012.